JPL D-7669, Part 2

Planetary Data System Standards Reference

December 16, 2010
Version 4.0.2 (beta)

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

PDS4 build 1b, based on:

Information Model Specification
Version 0.2.0.0.d
generated Mon Dec 13 20:27:44 PST 2010

schema files
built from the PDS4 ontology model V0.2.0.0.d
generated Mon Dec 13 20:27:43 PST 2010

Protegé ".pont" file
version 3.3.1
build 430
generated Mon Dec 13 15:16:09 PST 2010

Contents

PDS Standards Reference Change Log

Version	**Section**	**Change**
4.0.1	Appendix A	Added examples for Image_Grayscale

Chapter 1

Introduction

The Planetary Data System (PDS) Standards Reference is a complete specification for version 4 of the PDS standards. These standards are used to design data storage formats and encode descriptive labels for data stored in the PDS.

Note that version 4 of the PDS standards is *not* backwards compatible with version 3. All version 4 data can be described using version 3 labels, but the converse is not true.

This document *does not* provide a formal definition of the grammar of the eXtensible Markup Language (XML), which is used to encode the PDS4 standards; guidance on this aspect of PDS labels is provided in the Data Provider's Handbook (DPH).

1.1 PDS Data Policy

Only data that fully comply with PDS standards will be published as "Conforms to PDS Standards". When the PDS assists in the preparation of data published in a non-compliant format, PDS participation shall be acknowledged with a statement such as "funded by PDS". The PDS Management Council makes decisions on compliance waivers. Non-compliant data sets will only be incorporated into the PDS archives as "safed" data. (Do we wish to drop the statements about waivers?)

1.2 Purpose

The purpose of this document is to serve as a reference document detailing PDS standards used in the preparation of PDS compliant data. This document is to be used in conjuction with other PDS publications:

- *Archive Preparation Guide* - a brief overview of the archiving process (Is this being replaced by the DPH?)
- *Data Provider's Handbook* - an introduction and basic "how to" manual for archiving data with the PDS
- *PDS Policies and Processes* (Do we have a formal document name for this?)
- *Planetary Science Data Dictionary* - contains definitions of the standard classes and attributes used to describe PDS data
- *Proposer's Archiving Guide* - provides basic information on the archiving process (including PDS expectations) to scientists proposing for National Aeronautics and Space Administration (NASA) planetary programs
- *SPICE Archive Preparation Guide (PDS4 Version)* - describes the structure and contents of the PDS SPICE archive sets produced by Navigation and Ancillary Information Facility (NAIF) and the process that NAIF follows to put them together
- tutorials - contact the PDS Operator (pds_operator@jpl.nasa.gov) for more information on available tutorials

1.3 Scope

The information included here constitutes Version 4.0 of the Planetary Data System data preparation standards for producing archive quality data sets. This document covers the conceptual composition of an archive, its physical layout, and the current technology standards used for implementing the data and meta-data.

1.4 Audience

This document is intended primarily to serve the community of scientists and engineers responsible for preparing planetary science data sets for submission to the PDS. These include restored data from the era prior to PDS or from earlier versions of the PDS standards, mission data from active and future planetary missions, and data from earth-based sites. The audience includes personnel at PDS discipline and data nodes, mission principal investigators, and ground data system engineers. This document is intended for use by those people already somewhat familiar with the process of archiving data with the PDS. (Those new to the PDS should first read the DPH.)

1.5 Document Organization

The first section of this document, “Introduction”, provides introductory material and citations of other reference documents. The second section, “Conceptual Composition of an Archive” discusses the constituent components of an archive. The third section, “Physical Composition of an Archive” discusses data format and organization. The fourth section, “Implementation Conventions”, discusses the labeling standards employed by the PDS for representation of meta-data and file descriptions. The appendices provide a detailed description of each of the user classes used to describe PDS data.

1.6 Other Reference Documents

The following references are cited in this document:

- Batson, R. M., (1987) “Digital Cartography of the Planets: its Status and Future”, *Photogrammetric Engineering & Remote Sensing 53*, 1211-1218.

- Davies, M.E., et al. (1991) “Report of the International Astronomical Union (IAU)/International Association of Geodesy (IAG)/Committee on Space Research (COSPAR) Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 1991”, *Celestial Mechanics, 53*, 377-397.

- Greeley, R. and Batson, R.M. (1990) *Planetary Mapping*, Cambridge University Press, Cambridge, 296p.

- *Guide on Data Entity Naming Conventions*, National Bureau of Standards (NBS) Special Publication 500-149.

- *PDS 2010 Operations Concept*, Jet Propulsion Laboratory (JPL) D-#### (Available from the PDS).
- Planetary Science Data Dictionary, JPL D-7116 Rev D, July 15, 1996, (Available from the PDS).
- *Planetary Data System Data Preparation Workbook Version 3.1*, JPL D-7669 Part 1, February 17, 1995, (Available from the PDS)
- *Issues and Recommendations Associated with Distributed Computation and Data Management Systems for the Space Sciences*, National Academy Press, Washington, DC, 111p.

1.7 External Standards

External standards which apply to the content of this document:

What about ASCII and UTF-8 standards? – EDR

Consultative Committee for Space Data Systems (CCSDS):

- CCSDS 641.0-B-2 Parameter Value Language Specification (CCSD0006 and CCSD0008) (also available as ISO 14961:2002)

Institute of Electrical and Electronics Engineers (IEEE):

- IEEE 754-2008 *Standard for Binary Floating-Point Arithmetic*

International Standards Organization (ISO):

- ISO 646:1991 *ISO 7-bit coded character set for information interchange*
- ISO 8601:1988 *Data Element and Interchange Formats – Representations of Dates and Times* (Shouldn't this be 8601:2004? - EDR)
- International Standards Organization / International Electrotechnical Commission (ISO/IEC) 11179-3:2003 *Metadata registries (MDR) – Part 3: Registry metamodel and basic attributes*
- ISO/IEC 11404:2007 *General-Purpose Datatypes (GPD)*
- ISO 14721:2003 *Open archival information system – Reference model*

- International Standards Organization / Technical Standard (ISO/TS) 15000-3:2004 *electronic business eXtensible Markup Language (ebXML) – Part 3: Registry information model specification (ebRIM)*
- ISO/TS 15000-4:2004 *ebXML – Part 4: Registry services specification (ebRS)*

National Institute of Standards and Technology (NIST):

- NIST Special Publication 330 *The International System of Units (SI)*, United States version of the English text of the eighth edition (2006), Issued March 2008

World Wide Web Consortium (W3C):

- *XML 1.1* 2nd ed., August 16, 2006
- *XML Schema Part 0: Primer* 2nd ed., October 28, 2004
- *XML Schema Part 1: Structures* 2nd ed., October 28, 2004
- *XML Schema Part 2: Datatypes* 2nd ed., October 28, 2004

1.8 Online Document Availability

All PDS documents pertaining to archive preparation are available online. Information on accessing these references may be found on the PDS website at the following Uniform Resource Locator (URL):

http://pds.nasa.gov

To obtain a copy of these documents or for questions concerning these documents, contact the PDS Operator (at pds_operator@jpl.nasa.gov, 818-393-7165) or any PDS data engineer.

The examples provided throughout this document are based on both existing and planned PDS archive products, modified to reflect the current version of the PDS Standards. Discipline-specific extensions to the high-level classes defined in this document are created and augmented from time to time, as user community need arises. To check the current status of any discipline- or mission-specific class definition, consult a PDS data engineer.

Additional schemata and examples are available online at the following URL:

http://pds.nasa.gov/repository/pds4/SCHEMA/

Part I

Archive Structure Standards

Chapter 2

Archive Organization and Directory Contents

Data providers must deliver data organized in the following directory structure. PDS discipline nodes may or may not choose to retain that structure in their online repositories.

Data delivered to the PDS shall be assigned to directories based on product type.[1] The following sections describe the contents of the root directory, followed by the contents of the subdirectories in alphabetical order. The root directory is the top-level directory of a volume.

2.1 Root Directory

The root directory is the top level directory of an archive bundle. It contains all the other directories described in the following sections.

File or Directory Name	Class	Cardinality	Name Fixed?
root	–	1	not fixed
aareadme.xml[2]	Product_Document	1	fixed
bundle.xml	Product_Bundle	1	fixed
errata.xml[2]	Product_Document	0..1	fixed
about	–	1	fixed
browse	–	0..1	fixed

[2]membership is unresolved – possibly document collection?

[3]Note: DPH makes these optional – need to resolve.

[1]The one exception is the **about** directory, which may contain products of multiple different types.

File or Directory Name	Class	Cardinality	Name Fixed?
calibration	–	0..1	fixed
context	–	1[3]	fixed
data	–	1	fixed
document	–	1[3]	fixed
geometry	–	0..1	fixed
SPICE	–	0..1	fixed
xml_schema	–	1	fixed

[2] membership is unresolved – possibly document collection?

[3] Note: DPH makes these optional – need to resolve.

The aareadme.xml file contains a brief overview of the contents and organization of the associated archive, general instructions for its use, and contact information. The name has been chosen so that it will be listed first in an alphabetical directory listing.

The aareadme shall be implemented using the Product_Document class, and shall contain an Identification_Area_Document class with the product_subclass attribute set to "AAREADME". There must be one, and only one, of these files included in each bundle. The filename, "aareadme.xml", is fixed.

The bundle.xml file identifies all of the component collections of which the archive bundle is comprised.

The bundle product shall be implemented using the Product_Bundle class. There must be one, and only one, of these files included in each bundle. The filename, "bundle.xml", is fixed.

The errata.xml file identifies and describes errors and/or anomalies found in the archive.

The errata shall be implemented using the Product_Document class, and shall contain an Identification_Area_Document class with the product_subclass attribute set to "ERRATA". There must be one, and only one, of these files included in each bundle. The filename, "errata.xml", is fixed.

2.2 About Directory

The about directory contains supplemental meta-data catalogs, database dumps, indices, or spreadsheets deemed by the data provider to be useful to the interpretation of the data in the archive. Any updates made to standard product label values after the archive is ingested may be included in the about directory in tabular form. Examples of the types of information contained in the about directory include:

modification history
errata
updates
tables anaglyph pairs
footprint files
database dumps

File or Directory Name	Class	Cardinality	Name Fixed?
about	–	1	fixed
collection_misc.xml	Collection_Miscellaneous	1	fixed
collection_misc_manifest.tab	–	1	fixed
about_products	–	1[4]	fixed
⋮	(as appropriate)	⋮	⋮

[4]Note that DPH permits multiple **about_products** directories; I'm not sure why this is necessary, rather than having multiple subdirectories underneath it. - EDR

The about collection shall be implemented using a Collection_Miscellaneous class.

2.3 Browse Directory

The browse directory contains one or more browse collections. Each browse collection contains "quick-look" products designed to facilitate use of the archive. Browse products and the browse directory are optional.[5]

File or Directory Name	Class	Cardinality	Name Fixed?
browse	–	0..1	fixed
collection_browse.xml	Collection_Browse	1..*	fixed[6]
collection_browse_manifest.tab	–	1..*	fixed[6]
browse_products	–	1..*	fixed[6]
browse_product1.xml	(as appropriate)	1..*	not fixed
browse_product1.jpg	–	1..*	not fixed
⋮	⋮	⋮	⋮

[6]In the case of multiple browse collections these fixed names may be modified by adding a prefix to distinguish among the collections. For example: edr_collection_browse.xml and rdr_collection_browse.xml.

[5]Note that there are two alternative methods for including browse objects in an archive. The first is described in this section, namely, to include stand-alone browse products in a browse collection stored in a browse directory. The second method is to store the browse object alongside the primary data object in the science data product. (An example would be to stored a low-resolution JPEG formatted image in a separate file, but in the same directory, as a full-resolution raster-formatted image, and describe them both as a single product using a common PDS label. The latter primarily works in the case of a one-to-one correspondence between browse objects and primary data objects.

For each browse collection present in the archive, there shall be one collection_browse_manifest.tab file, one collection_browse.xml file and one browse_products sub-directory. If only one browse collection is present, these names are fixed. If more than one browse collection is present, these names may be prefixed with an additional identifying word terminated in an underscore, to distinguish the collections. In all such cases, the collection label, manifest, and directory must all be clearly identified with one another by utilizing the same prefix.

The browse collection shall be implemented using the Collection_Browse class.

If appropriate, each browse_products directory may contain multiple sub-directories. The structure of these sub-directories is at the discretion of the data provider. A structure that parallels the structure of the data subdirectories is frequently utilized.

2.4 Calibration Directory

The calibration directory contains calibration data and files necessary for the calibration of science data products.

File or Directory Name	Class	Cardinality	Name Fixed?
calibration	–	0..1	fixed
collection_calibration.xml	Collection_Calibration	1..*[7]	fixed
collection_calibration_manifest.tab	–	1..*[7]	fixed
calibration_products	–	1..*[7]	fixed
calibration_product1.xml	(as appropriate)	1..*	not fixed
calibration_product1.tab	–	1..*	not fixed
⋮	⋮	⋮	⋮

[7]Can there be multiple collections in the **calibration** directory? DPH says "no"; I suspect we might want the possibility. - EDR

TBD - Will not be completed for the first release.

2.5 Context Directory

The context directory contains all context products associated with the archive. These are the products identified in the Product_Cross_Reference_Area of the data products in the archive.

File or Directory Name	Class	Cardinality	Name Fixed?
context	–	1	fixed
collection_context.xml	Collection_Context	1	fixed
collection_context_manifest.tab	–	1	fixed
context_products	–	1	fixed
context_inst.xml	Product_Instrument	0..1[8]	fixed
context_insthost.xml	Product_Instrument_Host	0..1[8]	fixed
context_investigation.xml	Product_Investigation	0..1[8]	fixed
context_node.xml	Product_Node	0..1[8]	fixed
context_personnel.xml	Product_PDS_Affiliate or Product_PDS_Guest[9]	0..1[8]	fixed
context_publication.xml	Product_Citation[9]	0..1[8]	fixed
context_resource.xml	Product_Resource	0..1[8]	fixed
context_target.xml	Product_Target	0..1[8]	fixed

[8] We need to settle on which of these context files are required and which are optional. - EDR

[9] I'm not certain exactly how this is supposed to be implemented. Will probably need to be updated. - EDR

The context products are members of a context collection, the primary copy of which is archived as part of the PDS system bundle. Is this correct? Thus, in the context of an archive bundle containing standard data products, the context collection is a *secondary* bundle. The context collection shall be implemented using the Collection_Context class.

2.6 Data Directory

The data directory contains one or more data collections. Each data collection contains standard data products.

For regular science archives, the data directory is required. For special archives (like a SPICE bundle or a system bundle), the directory is optional.

File or Directory Name	Class	Cardinality	Name Fixed?
data	–	0..1	fixed
collection_data.xml	Collection_Data	1..*	fixed[9]
collection_data_manifest.tab	–	1..*	fixed[9]
data_products	–	1..*	fixed[9]
subdir1	–	1..*	not fixed
data_product1.xml	(as appropriate)	1..*	not fixed
data_product1.tab	–	1..*	not fixed

[9] In the case of multiple data collections these fixed names may be modified by adding a prefix to distinguish among the collections. For example: edr_collection_data.xml and rdr_collection_data.xml labels would correspond to edr_data_products and rdr_data_products directories.

File or Directory Name	Class	Cardinality	Name Fixed?
⋮	⋮	⋮	⋮
subdir2	–	1..*	not fixed
data_productn.xml	(as appropriate)	1..*	not fixed
data_productn.tab	–	1..*	not fixed
⋮	⋮	⋮	⋮

[9] In the case of multiple data collections these fixed names may be modified by adding a prefix to distinguish among the collections. For example: edr_collection_data.xml and rdr_collection_data.xml labels would correspond to edr_data_products and rdr_data_products directories.

There shall be one collection This directory may be further sub-divided to prevent over-crowding of directories and facilitate archive organization.

2.7 Document Directory

The document directory ...

2.8 Geometry Directory

The geometry directory contains non-Spacecraft, Planet, Instrument, C-matrix (pointing), and Events kernels (SPICE) geometry files (for example, Supplementary Experiment Data Record (SEDR)).

(What is the relationship between cartographic information and the geometry directory?)

2.9 SPICE Kernel Directory

(all SPICE data inclu

(same directory structure as for SPICE Bundles)

The ”spice” directory in a PDS bundle can contain individual SPICE kernel files, their XML labels, and XML labels describing primary and/or secondary SPICE collections.

If the PDS bundle contains any SPICE kernel files, these kernel files and their XML labels must be placed in the following subdirectories based on the kernel type:

ck	CK files (spacecraft and instrument orientation data)
dbk	DBK files (databases in SPICE format)
dsk	DSK files (digital shape data for natural bodies)
ek	EK files (events information)
fk	FK files (reference frames definitions)
ik	IK files (instrument parameters and FOV definitions)
lsk	LSK files (leapsecond information)
mk	MK files (meta-kernels listing kernels to be used together)
pck	PCK files (natural body rotation and size/shape constants)
sclk	SCLK files (spacecraft clock correlation data)
spk	SPK files (trajectory and ephemeris data)

All SPICE kernel files provided in the ”spice” directory in the Bundle must be members of a single primary SPICE collection. The XML label describing this SPICE collection must reside in the ”spice” directory.

The ”spice” directory in a Bundle may also contain additional XML labels for secondary SPICE collections, pointing to the kernels in the Bundle and/or kernels in other Bundles or SPICE collections provided in other Bundles.

If SPICE kernel files of a particular type are not included in the Bundle, the subdirectory for that kernel type must be omitted.

If no SPICE kernel files are included in the Bundle, all subdirectories and the primary collection XML label must be omitted. In this case the ”spice” directory must contain only the XML labels for secondary SPICE collections.

If no SPICE kernel files and primary and secondary SPICE collections are included in the Bundle, the ”spice” directory must not be present in the Bundle.

2.10 XML Schema Directory

The XML schema collection contains the XML schema files included in or referenced by XML labels in the archive.

There is one XML schema collection per archive.

collection_xml_schema.xml

collection_xml_schema_manifest.tab

xml_schema_products

There is one required subdirectory to the XML schema directory. It contains

Chapter 3

Data Objects and Products

This chapter still needs a lot of work.

PDS products consist of one or more objects and a detached product label. Product labels are required to describe the contents and structure of the objects. PDS product labels are formatted in XML.

There is always a one-to-one correspondence between a PDS product and its label.

As has been stated elsewhere, in Planetary Data System, standards version 4 (PDS4) everything is considered to be a product. Thus, in addition to traditional data products such as images, tables, and histograms, missions and spacecraft are now also treated as products. There are three

3.1 Terminology: Products, Objects, Classes, and Files

Within the data model, all *things* are treated as one of three types of objects: physical, digital, or conceptual.

Physical objects are those which are tangible. Planets, spacecraft, and instruments are examples of physical objects which are tracked in the PDS.

Digital objects make up the bulk of the PDS holdings. These are the types of things one traditionally thinks of as planetary science data: digital files containing raster images, binary or character tables, documents, and software.

Conceptual objects are intangibles: missions, observing campaigns, and academic institutions are all examples of this type of object.

An object, accompanied by a description of that object, is called a *tagged digital object* (in the case of digital objects) or a *tagged nondigital object* (in the case of physical and conceptual objects).

One or more tagged objects, grouped together and described with a single label, constitute a *product*. This is true for all three types of objects mentioned above. Thus a planet and the description of that planet is considered to be a product no less than an image and the description of that image.

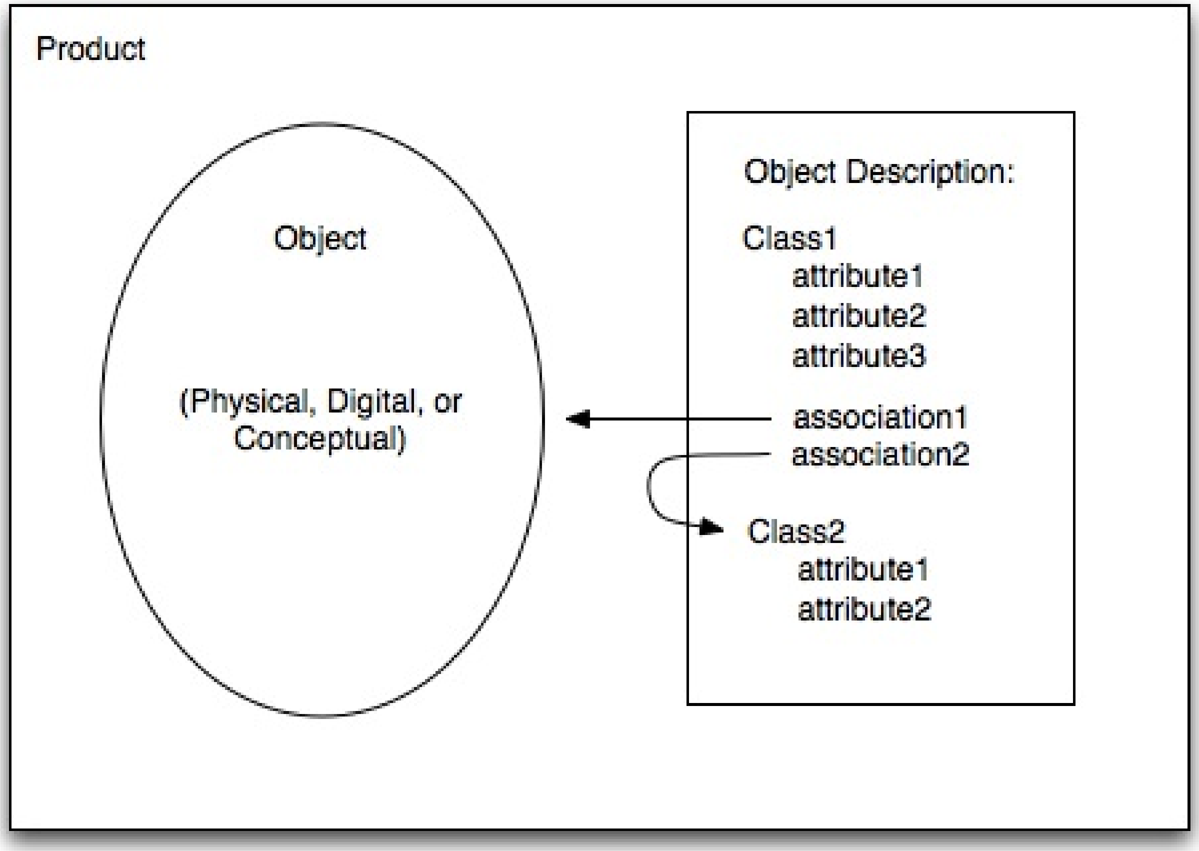
Furthermore, a *list* of products (accompanied by a description of the list) is also considered to be a product. Thus one may identify and track logical sets of products.

Within the PDS archive, three types of products are recognized: *standard* products, *collections*, and *bundles*. Each of these are described in more detail in the following sections.

Each type of object recognized by the PDS is described in the data model with a *class*. Thus the Mission class is used to describe a mission (conceptual) object.

Classes contain *attributes* and *associations*. The attributes are used to provide information about the objects. For example, the Mission class contains start_date and stop_date attributes that provide information about a particular mission object.

Associations describe relationships between the class and other entities, be they objects or other classes.



3.2 Standard Products

The lowest level product in the PDS is referred to as a standard product. A standard product comprises one or more objects constituting (typically) a single observation or a single document.

standard product can only belong to one primary collection primary collections may only reference lidvids

3.3 Collection Products

The collection manifest table shall consist of one record for each product in the collection. Each record shall contain a product LIDVID (see 7.1.1), followed by the file_specification_name for the product's label.

3.4 Bundle Products

The highest level product is referred to as a *bundle*. Like collections, bundles consist of a list of references to products; however in this case, the referenced products may themselves be collections. A bundle identifies all of the collections and standard products necessary to perform useful science analysis on the data contained therein.

Chapter 4

Labels

PDS product labels are required for describing the contents and format of each individual product within an archive. Labels are populated using a standard set of classes, elements, and standard values for the elements. These classes, elements, and standard values are defined in the Planetary Science Data Dictionary (PSDD). Appendices A through C of this document provide general descriptions and examples of the the classes and their data element used in labeling PDS products.

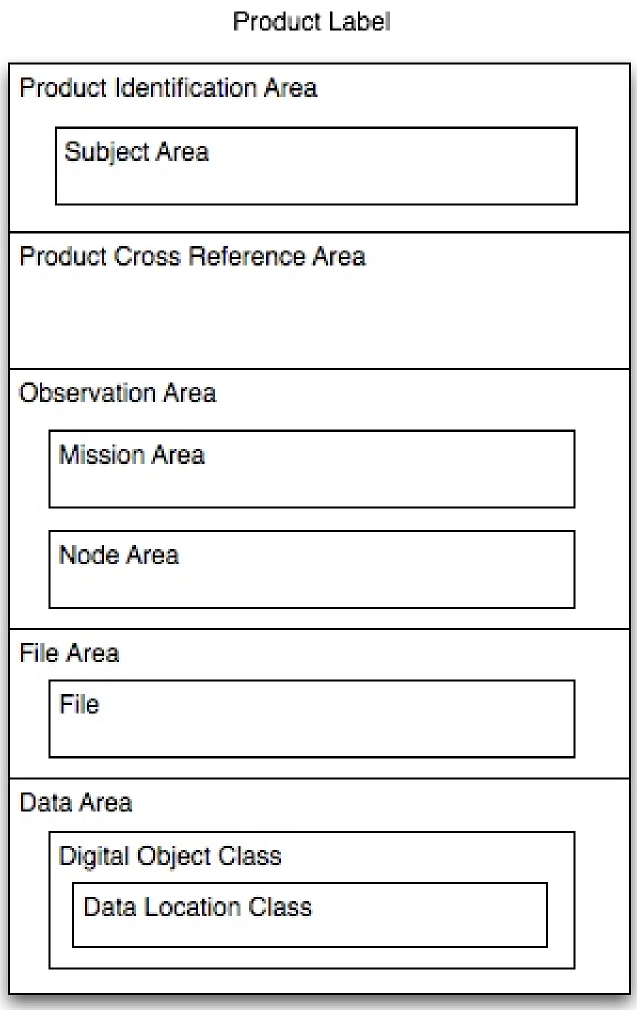
Under the PDS4 standard, all product labels are detached from the data files they describe. There is one label file for every product. Each label file may describe one or more data files.

For details on the specific grammar used to implement the labels, see section IV of this document, *Implementation Conventions*.

PDS product labels have a general structure that is used for the vast majority of products.

- Identification Area
- Cross Reference Area
- Observation Area
- File Area
- Data Area
- Data Area Alternate

Figure x.x shows how this general structure appears in a product label.


Product Label
Product Identification Area
Subject Area
Product Cross Reference Area
Observation Area
Mission Area
Node Area
File Area
File
Data Area
Digital Object Class
Data Location Class

4.1 The Product_Identification Area

4.1.1 The Subject Area

4.2 The Product_Cross_Reference Area

4.3 The Observation Area

4.4 The File Area

The "File_Area" of a label is used to describe the system files containing the digital objects described in the "Data_Area". It must contain a distinct File class for each file containing data for this product.

Each File class must be described by a local_identifier, unique within the product label. This identifier will be used in the Data_Location class (described below) to tie together a Digital Object class with the digital data it describes. Local_identifiers may be re-used from one label to another within a collection.

4.5 The Data Area

4.5.1 The Digital Object Classes

4.5.1.1 The Data_Location Class

Chapter 5

Fundamental Data Structures

There are four fundamental data structures that may be used for archiving data in the PDS. All data products delivered to the PDS must be constructed from one or more of these structures. These four fundamental structures are described using four base classes: Table_Base (used to describe heterogeneous repeating records of scalars), Array_Base (used for homogeneous n-dimensional arrays of scalars), Parsable_Byte_Stream, and Encoded_Byte_Stream. All other digital object classes in the PDS are derived from one of these four.

5.1 Table Base

In the PDS, column data may be stored in either of two different structures, either a fixed-width "table", or a variable-length "spreadsheet" (delimited stream). The latter is described in section **??**, "Unencoded Stream Base".

Conceptually, tabular data files consist of a series of named columns containing both data locations and data values. The data may consist of both numbers and text strings.

The TABLE object is a uniform collection of rows containing American Standard Code for Information Interchange (ASCII) or binary values stored in columns. The INTERCHANGE_FORMAT keyword is used to distinguish between TABLEs containing only ASCII columns and those containing binary data. The rows and columns of the TABLE object provide a natural correspondence to the records and fields often defined in interface specifications for existing data products. Each field is defined as a fixed-width COLUMN object; the value of the COLUMNS keyword is the total number of COLUMN objects defined in the label. All TABLE objects must have fixed-width records.

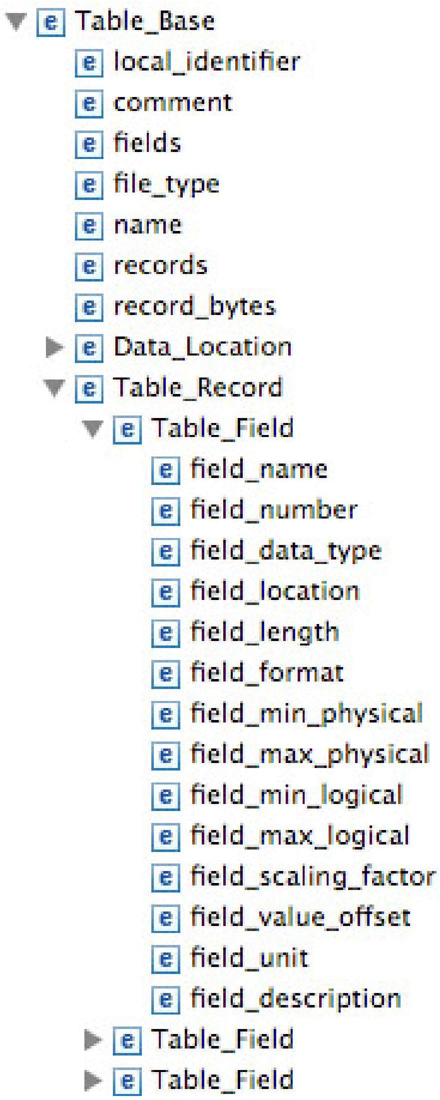


Figure 5.1: Table_Base

Physically, the data are stored as a sequence of repeating records where each record is terminated by both the CR and LF characters (ASCII characters xxx and yyy, respectively). Since both the column lengths and the record lengths are fixed, column values are identifiable by position alone. However, column delimiters may optionally be included.

The data may be represented in any of binary, ASCII, or Unicode Transformation Format (UTF)-8 values.

(include picture here)

5.2 Array Base

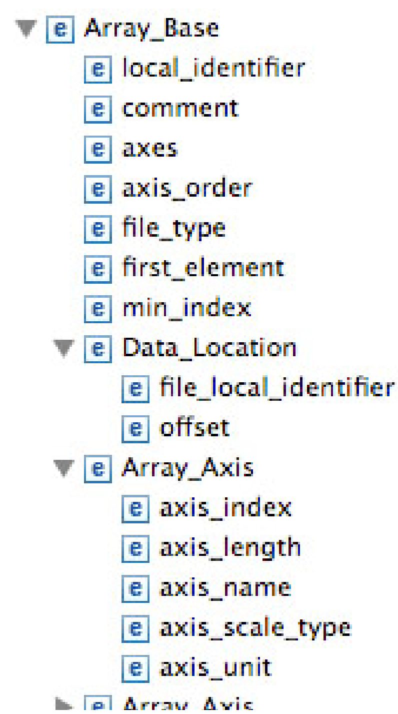
The second of the four basic structures consists solely of homogeneous pixels in fixed-length arrays in any number of dimensions, although two- and three-dimensional data are the most common. These arrays are described using a subclass of the Array_Base class, which has two associated classes (besides the Data_Location class), the Array_Axis and Array_Element classes.

The structure of the homogeneous pixels or "elements" in an array is described using the Array_Element class. The amount of physical storage space required for each element is describe by the element_bytes attribute. The storage order and interpretation of the bytes comprising each element is indicated by the element_type attribute, which typically has values like "SignedMSB2" or "IEEE754Double". (See the Planetary Science Data Dictionary for a complete listing of valid data types.)

The physical interpretation of the element values is described using the remaining three attributes. The element_unit attribute describes the ...??? The element_scaling_factor and element_offset_value attributes provide the numbers by which an element value must be transformed, using the following formula:

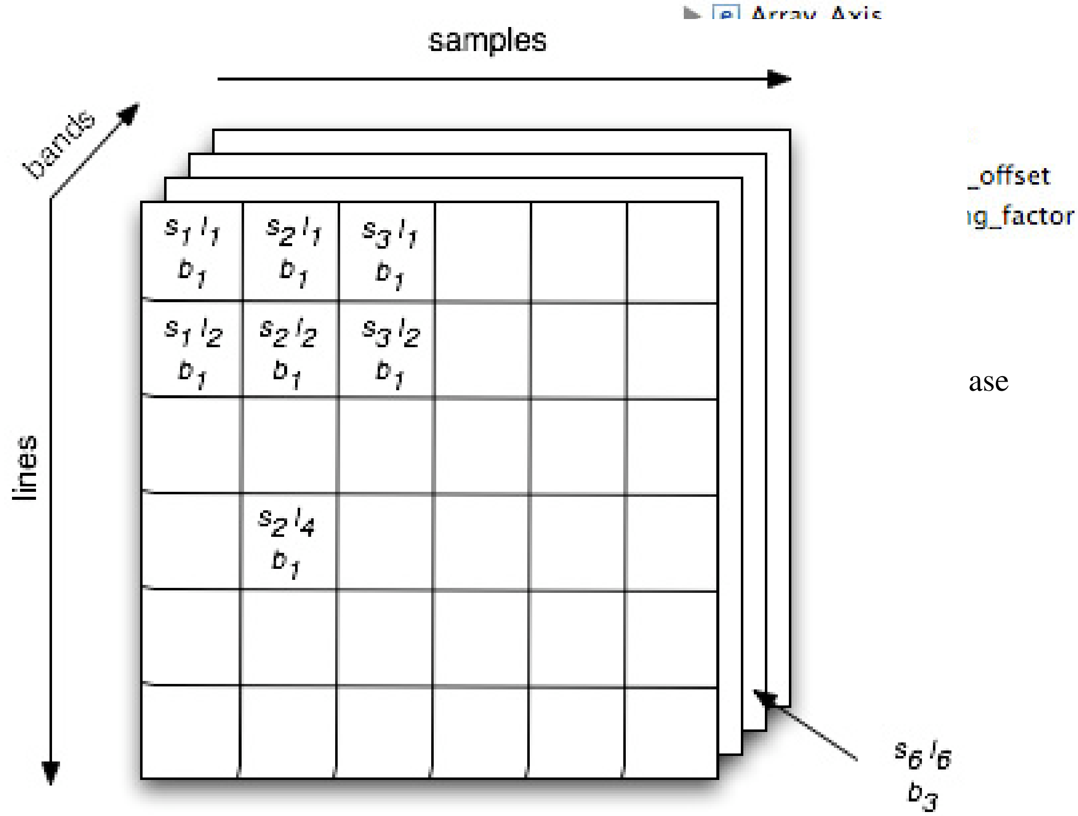
true value = (stored value x scaling factor) + offset value

The characteristics of each array axis are modeled using an Array_Axis class (there must be one Array_Axis class present in the label for each dimension of the array). The axis_length attribute provides the number of elements in the array along each particular axis. The axis_name attribute unambiguously labels each axis, and may have a value as simple as "line", "sample", or "spectral band". (Axis_unit and axis_scale_type still have significant problems.)



Finally, the Array_Base class, or one of its subclasses, wraps the whole thing together:

The axis_order attribute specifies the order in which the elements of the multi-dimensional array are stored. The PDS requires that the data be stored in a fastest-varying-pixel-first order. In other words, for the following conceptual representation of a 3-dimensional array:



the data stream must contain the pixels in the following order:

$s_1l_1b_1, s_2l_1b_1, s_3l_1b_1...s_1l_2b_1, s_2l_2b_1, s_3l_2b_1...s_1l_1b_2, s_2l_1b_2, s_3l_1b_2...$

This ordering of the elements is indicated by specifying an axes_order of ”FIRST_INDEX_FASTEST”. (I am not convinced that we have yet completely resolved the issue of tying together the axis order with the named Array_Axis classes. – EDR)

5.3 Parsable Byte Stream

XML / Hypertext Markup Language (HTML) / spreadsheet xhtml text files telemetry stream format files - conceivably

Alternatively, data may be stored in a variable-length structure. The PDS refers to these structures as ”spreadsheets”.

5.4 Encoded Byte Stream

The encoded stream base structure in the PDS is a byte stream that may only be interpreted after it has been ”decoded”, according to some well known standard. ”Encoded” data may include data that has been compressed and needs to be decompressed before interpretation. Alternatively, it may include data encoded in some form of binary format, such as the Portable Document Format (PDF).

In order to interpret an encoded stream base, reading software would need to determine the value of the external_standard_id attribute and access the referenced standard for information on how to parse the byte stream. It is PDS policy that only publicly available, open source, widely accepted standards may be used for the encoding of data within the PDS.

Chapter 6

Data Storage Types

6.1 Attribute Data Types

Attribute data types are used to classify attributes in class descriptions (i.e., in product labels).

Value	Description	Values
ASCII_AnyURI	A URI or its subclasses URN and URL (See section 7.2.4 for details.)	An ASCII string in URI format. Limit 255 characters.
ASCII_Boolean_TF	True / False indicator	`T` or `F`[1] Limit 1 character.
ASCII_Date[2]	A date string in either Day Of Year (DOY) or Year Month Day (YMD) format.	Date value in either of the following two forms: `yyyy-doy` `yyyy-mm-dd`.
ASCII_Date_DOY	A date string in DOY format.	Date value of the form: `yyyy-doy`.

[1] Are other boolean values permitted? – EDR

[2] Note: this data_type is not currently in the information model. However, given that we have a date/time data_type that permits either date string format, shouldn't we have a generic date data_type as well? – EDR

[3] Not sure I understand the necessity for both this and ASCII_Date_Time_YMD – EDR

[4] Information model specifies limit of 1 character and max value of 1; I believe these to be erroneous. – EDR

[5] Information model shows three colons between LID and VID; need to fix. Also need to verify version id format is consistent with that agreed upon by DDWG. – EDR

[6] Information model shows minimum_characters = 1; should be 32. – EDR

[7] Should the max value be the same for ASCII_Integer and ASCII_NonNegative_Integer? – EDR

[8] Do we really mean to have no limits on this data type? – EDR

[9] Need to verify version id format is consistent with that agreed upon by DDWG. – EDR

Value	Description	Values
ASCII_Date_YMD	A date string in YMD format.	Date value of the form: `yyyy-mm-dd`.
ASCII_Date_Time	A date/time string where the date may be in either DOY or YMD format.	Date/time value in either of the following two forms: `yyyy-doyThh:mm:ss[.fff]` `yyyy-mm-ddThh:mm:ss[.fff]`
ASCII_Date_Time_DOY	A date/time string where the date is in DOY format.	Date/time value of the form: `yyyy-doyThh:mm:ss[.fff]`
ASCII_Date_Time_UTC[3]	A date/time string in Coordinated Universal Time (UTC) format.	Date/time value of the form: `yyyy-mm-ddThh:mm:ss[.fff]Z`
ASCII_Date_Time_YMD	A date/time string where the date is in YMD format.	Date/time value of the form: `yyyy-mm-ddThh:mm:ss[.fff]` Limited to ASCII characters
ASCII_Directory_Path_Name	A system directory path in UNIX format.	ASCII string of the form: `dir1/dir2/` Limit 255 characters.
ASCII_DOI	A Digital Object Identifier (DOI). (See section 7.2.3 for details.)	ASCII string of the form: `nn.nnnn/nnn` Limit 255 characters.
ASCII_File_Name	A system file name. (See section 7.4 for details.)	ASCII string of the form: `file_name.file_ext` Limit 255 characters.
ASCII_File_Specification_Name	A system file including directory path, file name, and file extension in UNIX format.	ASCII string of the form: `dir1/dir2/file_name.ext` Limit 255 characters.
ASCII_Identifier	A PDS identifier. (See section 7.1 for details.)	ASCII string beginning with an alphabetic character. Limit 100 characters.

[1] Are other boolean values permitted? – EDR

[2] Note: this data_type is not currently in the information model. However, given that we have a date/time data_type that permits either date string format, shouldn't we have a generic date data_type as well? – EDR

[3] Not sure I understand the necessity for both this and ASCII_Date_Time_YMD – EDR

[4] Information model specifies limit of 1 character and max value of 1; I believe these to be erroneous. – EDR

[5] Information model shows three colons between LID and VID; need to fix. Also need to verify version id format is consistent with that agreed upon by DDWG. – EDR

[6] Information model shows minimum_characters = 1; should be 32. – EDR

[7] Should the max value be the same for ASCII_Integer and ASCII_NonNegative_Integer? – EDR

[8] Do we really mean to have no limits on this data type? – EDR

[9] Need to verify version id format is consistent with that agreed upon by DDWG. – EDR

Value	Description	Values
ASCII_Integer	An ASCII character representation of a decimal integer.	ASCII number in the range -2147483648 to 2147483647.
ASCII_Integer_Binary	An ASCII representation of a number in binary format.	An ASCII string of 0s and 1s. Limit 32 characters.[4]
ASCII_Integer_Hex	An ASCII representation of a number in hexadecimal format.	An ASCII string consisting of the characters 0 through 9 and A through F. Limit 32 characters.[4]
ASCII_LID	A PDS logical identifier. (See section 7.1.1 for details.)	An ASCII string of the form `URN:NASA:PDS:xxxx` Limit 255 characters.
ASCII_LIDVID	A PDS versioned identifier (logical identifier plus version id). (See section 7.1.1 for details.)	An ASCII string of the form `URN:NASA:PDS:xxxx::M.n`[5]Limit 255 characters.
ASCII_MD5_Checksum	A 128-bit hash value calculated using the Message-Digest algorithm 5 (MD5).	An ASCII string consisting of the characters 0 through 9 and A through F. Must be exactly 32 characters in length.[6]
ASCII_NonNegative_Integer	An ASCII character representation of a decimal integer greater than or equal to zero.	ASCII number in the range 0 to 2147483647.[7]
ASCII_Real	An ASCII character representation of a real number.	ASCII number in the range of negative infinity to positive infinity.[8]

[1]Are other boolean values permitted? – EDR

[2]Note: this data_type is not currently in the information model. However, given that we have a date/time data_type that permits either date string format, shouldn't we have a generic date data_type as well? – EDR

[3]Not sure I understand the necessity for both this and ASCII_Date_Time_YMD – EDR

[4]Information model specifies limit of 1 character and max value of 1; I believe these to be erroneous. – EDR

[5]Information model shows three colons between LID and VID; need to fix. Also need to verify version id format is consistent with that agreed upon by DDWG. – EDR

[6]Information model shows minimum_characters = 1; should be 32. – EDR

[7]Should the max value be the same for ASCII_Integer and ASCII_NonNegative_Integer? – EDR

[8]Do we really mean to have no limits on this data type? – EDR

[9]Need to verify version id format is consistent with that agreed upon by DDWG. – EDR

Value	Description	Values
ASCII_Short_String_Collapsed	An ASCII-encoded text string of limited length with whitespace collapsed. (I.e. multiple spaces, new lines, tabs, and carriage returns are not significant.)	An ASCII string. Limit 255 characters.
ASCII_Short_String_Preserved	An ASCII-encoded text string of limited length with whitespace preserved. (I.e. multiple spaces, new lines, tabs, and carriage returns are significant.)	An ASCII string. Limit 255 characters.
ASCII_Text_Preserved	An ASCII-encoded text string of unlimited length with whitespace preserved.	An ASCII string.
ASCII_Time	An ASCII time string. May be used for local times.	ASCII string of the form: `hh:mm:ss[.fff]`. If UTC, `Z` should be appended.
ASCII_VID	A PDS version id. (See section 7.2.1 for details.)	An ASCII string of the form `M.m`, where `M` and `m` are both integers.[9]
UTF8_Short_String_Collapsed	A UTF-8 encoded text string of limited length with whitespace collapsed.	A UTF-8 string. Limit 255 characters.
UTF8_Short_String_Preserved	A UTF8-encoded text string of limited length with whitespace preserved.	A UTF-8 string. Limit 255 characters.
UTF8_Text_Preserved	A UTF8-encoded text string of unlimited length with whitespace preserved.	A UTF-8 string.

[1] Are other boolean values permitted? – EDR

[2] Note: this data_type is not currently in the information model. However, given that we have a date/time data_type that permits either date string format, shouldn't we have a generic date data_type as well? – EDR

[3] Not sure I understand the necessity for both this and ASCII_Date_Time_YMD – EDR

[4] Information model specifies limit of 1 character and max value of 1; I believe these to be erroneous. – EDR

[5] Information model shows three colons between LID and VID; need to fix. Also need to verify version id format is consistent with that agreed upon by DDWG. – EDR

[6] Information model shows minimum_characters = 1; should be 32. – EDR

[7] Should the max value be the same for ASCII_Integer and ASCII_NonNegative_Integer? – EDR

[8] Do we really mean to have no limits on this data type? – EDR

[9] Need to verify version id format is consistent with that agreed upon by DDWG. – EDR

6.2 Character Data Types

TBD – Do we still have a distinction between attribute data types and character data types? If not, what are we supposed to use to describe character data in table objects?

6.3 Binary Data Types

Need to verify the data types listed here against those in the Information Model. – EDR

6.3.1 Integers

6.3.1.1 Signed LSB Integers

This section describes signed integers stored in Least Significant Bit (LSB) order. In this section the following definitions apply:

b0 – b7 Arrangement of bytes as they appear when reading a file (e.g., read byte b0 first, then b1, b2 and b3, up through b7)

i-sign Integer sign bit – bit 7 in the highest order byte

i0 – i7 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7), in the following way:

8-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}
In i4, bits 0-7 represent 2^{32} through 2^{39}
In i5, bits 0-7 represent 2^{40} through 2^{47}
In i6, bits 0-7 represent 2^{48} through 2^{55}
In i7, bits 0-6 represent 2^{56} through 2^{62}

4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7

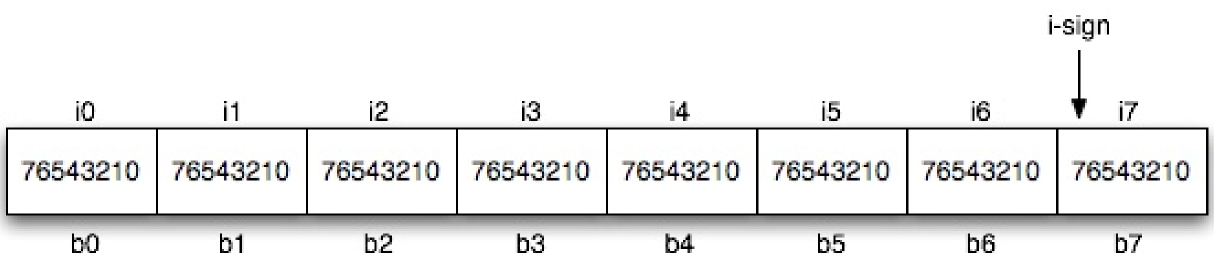
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-6 represent 2^{24} through 2^{30}

2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-6 represent 2^8 through 2^{14}

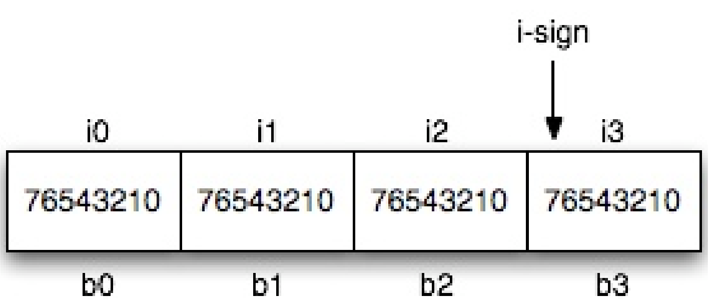
1-byte integers:
In i0, bits 0-6 represent 2^0 through 2^6

All negative values are represented in two's complement.

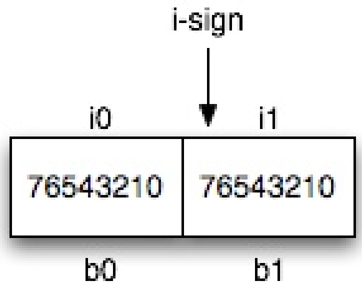
SignedLSB8



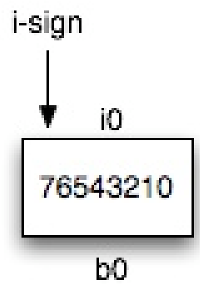
SignedLSB4



SignedLSB2



SignedByte



6.3.1.2 Unsigned LSB Integers

This section describes unsigned integers stored in Least Significant Byte first (LSB) format. In this section the following definitions apply:

b0 – b3 Arrangement of bytes as they appear when reading a file (e.g., read byte b0 first, then b1, b2 and b3)

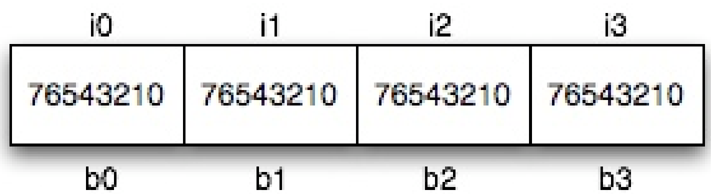
i0 – i3 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7), in the following way:

4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}

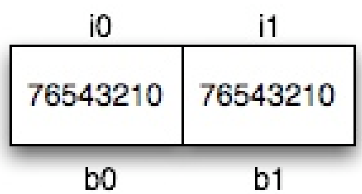
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}

1-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7

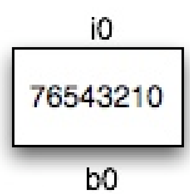
UnsignedLSB4



UnsignedLSB2



Unsigned Byte



6.3.1.3 Signed MSB Integers

This section describes the signed integers stored in Most Significant Byte first (MSB) order. In this section the following definitions apply:

b0 – b7 Arrangement of bytes as they appear when read from a file (e.g., read b0 first, then b1, b2, and b3 up through b7)

i-sign Integer sign bit – bit 7 in the highest order byte

i0 – i7 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7) in the following way:

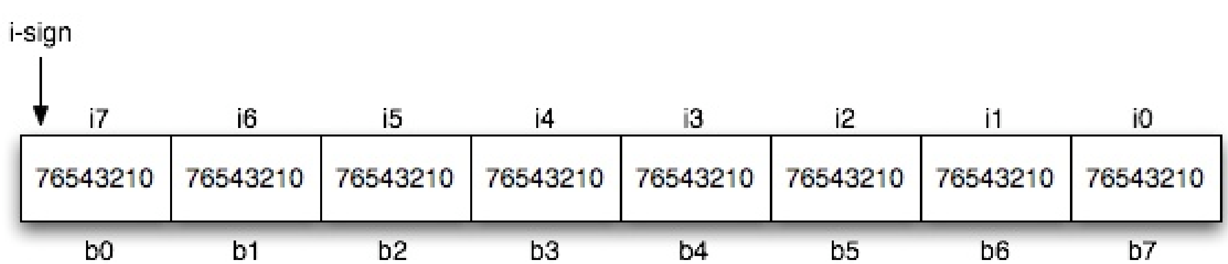
8-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}
In i4, bits 0-7 represent 2^{32} through 2^{39}
In i5, bits 0-7 represent 2^{40} through 2^{47}
In i6, bits 0-7 represent 2^{48} through 2^{55}
In i7, bits 0-6 represent 2^{56} through 2^{62}

4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-6 represent 2^{24} through 2^{30}

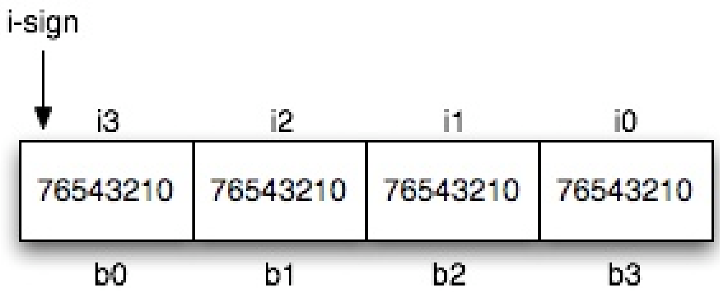
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-6 represent 2^8 through 2^{14}

1-byte integers:
In i0, bits 0-6 represent 2^0 through 2^6

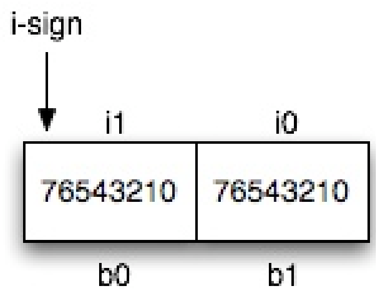
SignedMSB8



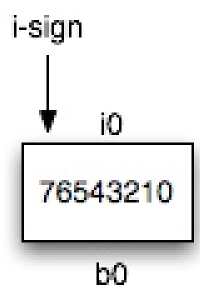
SignedMSB4



SignedMSB2



SignedByte



6.3.1.4 Unsigned MSB Integers

This section describes unsigned integers stored in Most Significant first (MSB) format. In this section the following definitions apply:

b0 – b3 Arrangement of bytes as they appear when read from a file (e.g., read b0 first, then b1, b2, and b3)

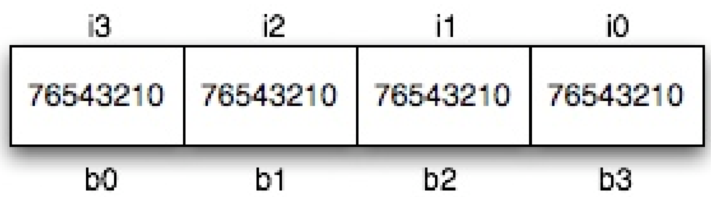
i0 – i3 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7) in the following way:

4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}

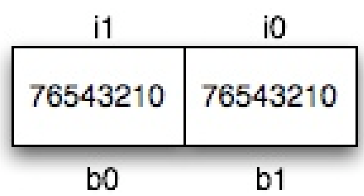
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}

1-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7

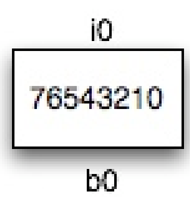
UnsignedMSB4



UnsignedMSB2



UnsignedByte



6.3.2 Reals

This section describes the internal format of IEEE-format floating-point numbers. In this section the following definitions apply:

b0 – b9 Arrangement of bytes as they appear when read from a file (e.g., read b0 first, then b1, b2, b3, etc.)

m-sign Mantissa sign bit

int-bit In 10-byte real format only, the integer part of the mantissa, assumed to be "1" in other formats, is explicitly indicated by this bit

e0 – e1 Arrangement of the portions of the bytes that make up the exponent, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = rightmost bit in the exponent part of the byte, highest value = leftmost bit in the exponent part of the byte) in the following way:

8-bytes (double precision):
In e0, bits 4-7 represent 2^0 through 2^3
In e1, bits 0-6 represent 2^4 through 2^{10}

Exponent bias = 1023

4-bytes (single precision):
In e0, bit 7 represents 2^0
In e1, bits 0-6 represent 2^1 through 2^7

Exponent bias = 127

m0 – m7 Arrangement of the portions of the bytes that make up the mantissa, from highest order fractions to the lowest order fraction. The order of the bits within each byte progresses from left to right, with each bit representing a fractional power of two, in the following way:

8-bytes (double precision):
In m0, bits 3-0 represent $1/2^1$ through $1/2^4$
In m1, bits 7-0 represent $1/2^5$ through $1/2^{12}$
In m2, bits 7-0 represent $1/2^{13}$ through $1/2^{20}$
In m3, bits 7-0 represent $1/2^{21}$ through $1/2^{28}$
In m4, bits 7-0 represent $1/2^{29}$ through $1/2^{36}$

In m5, bits 7-0 represent $1/2^{37}$ through $1/2^{44}$
In m6, bits 7-0 represent $1/2^{45}$ through $1/2^{52}$

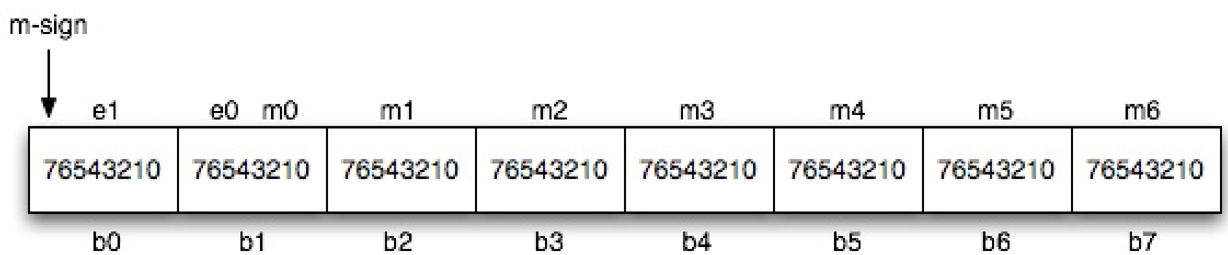
4-bytes (single precision):
In m0, bits 6-0 represent $1/2^{1}$ through $1/2^{7}$
In m1, bits 7-0 represent $1/2^{8}$ through $1/2^{15}$
In m2, bits 7-0 represent $1/2^{16}$ through $1/2^{23}$

The following representations all follow this format:

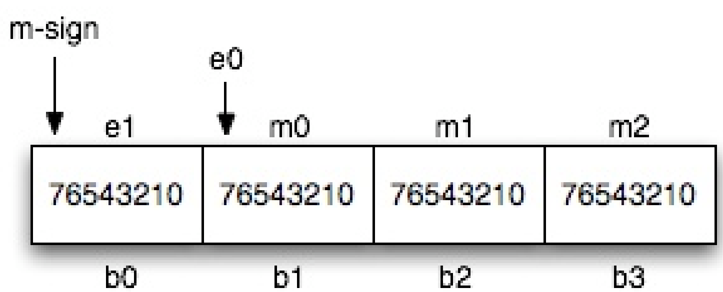
$$1.mantissa \text{ x } 2^{(exponent-bias)}$$

Note that the integer part ("1.") is implicit in all formats as described above. In all cases the exponent is stored as an unsigned, biased integer (that is, the stored exponent value - bias value = true exponent).

IEEE754Double



IEEE754Single



6.3.3 Complex

IEEE complex numbers consist of two IEEE_REAL format numbers of the same precision, contiguous in memory. The first number represents the real part and the second the imaginary part of the complex value.

6.4 Multi-Valued Data Types

Where do these definitions belong? One possibility is to toss out the section on vectors (since they're fairly simple to understand) and move the quaternion section to the chapter on Cartography??? This section is basically a placeholder until I come up with a better idea. – EDR.

6.4.1 Vectors

6.4.2 Quaternions

A quaternion is a four-component representation of a rotation matrix. This particular definition is focused on the PDS use of quaternions; one should refer to other sources for a more complete discourse on quaternion math.

A quaternion may be used to specify the rotation of one Cartesian reference frame–sometimes referred to as the base frame or the 'From' frame–into coincidence with a second Cartesian reference frame–sometimes referred to as the target reference frame or the 'To' frame. Unlike an Euler rotation where three sequential rotations about primary axes are used, a quaternion rotation is a single action, specified by a Cartesian vector used as the positive axis of the rotation (right hand rule) and the magnitude (an angle) of rotation about that axis.

The quaternion may be thought of as defining the instantaneous orientation–sometimes called 'pointing'–of a structure such as an instrument, antenna, solar array or spacecraft bus, given relative to a specified reference frame (the base frame), at an epoch of interest.

Perhaps of more use is the concept that a quaternion may be used to rotate an arbitrary Cartesian 3-vector defined in one reference frame (e.g. an instrument's reference frame) to an equivalent vector defined in another reference frame (e.g. the frame tied to a spacecraft or the J2000 inertial reference frame).

A quaternion has four components. One of the components is a scalar, a function of the angle of rotation (cosine of half the rotation angle), while the remaining three components are used to specify a vector, given in the base reference frame, about which the rotation will be made. In the PDS context a quaternion has a magnitude of one, and so may be treated as a unit quaternion.

In many cases a time tag (epoch) must be associated with the quaternion because the orientation varies over time. A time tag is not needed if the 'To' and 'From' frames have a fixed offset.

The QUATERNION_DESC element is always to be paired with the QUATERNION element, and will contain a complete description of the formation and rotational sense of the quaternion specified with the QUATERNION keyword, and the structure (organization of the four components) of the quaternion.

In the lingo of the NASA 'SPICE' ancillary information system a rotation matrix is synonymous with a C-matrix–that which may be obtained from a C-kernel. The SPICE Toolkit provides an assortment of routines that deal with quaternions. The SPICE system also provides information about specification of reference frames and time tags suitable for use with quaternions in the SPICE context. The NAIF Node of the PDS can provide additional documentation on quaternions in a spacecraft ancillary data context ('Rotations Required Reading' and 'SPICE Quaternion White Paper').

Chapter 7

Nomenclature Rules

7.1 Identifiers

7.1.1 Logical Identifiers vs. Versioned Identifiers

Each version of each product in the PDS has an identifier which is unique in the world. In the PDS, we refer to this as a *versioned identifier* or *lidvid*.

This identifier, minus the version information, is referred to as a *logical identifier* or *lid*. The logical identifier is used to denote all versions of a product collectively.

lidvid = lid "::" version id

Versioned

length limit on versioned identifiers is 255 characters; there is no specific limit on any individual component of these, but missions must plan well in advance to leave enough characters for the longest LIDVID (the product LIDVID), INCLUDING ALLOWING FOR HIGH LEVEL VERSION IDs!

stay well under the limit in order to accommodate the inevitable and unanticipated changes that will come down the road (higher sol numbers than expected due to extended missions, higher version ids

7.1.2 Product Identifiers

Product identifiers are used to register PDS products and must be **globally** unique.

product logical identifier (lid) = <collection logical identifier> “:” <specific identifier>
product versioned identifier (lidvid) = <product logical identifier> “::” <product version id>

specific identifier = a string, determined by the data provider, that is unique within the collection

Many data providers choose to use the file name for specific identifier. In this case, it shall consist only of the file name base of the label file (i.e., without the file name extension). If the file name base is not unique within the collection, information (such as the relative directory path) shall be pre-pended to it to make it unique.

Rules:

- Must be unique across the PDS.
- (character set is same as for file names)

7.1.3 Collection Identifiers

collection logical identifier (lid) = <bundle logical identifier> “:” <specific identifier>
collection versioned identifier (lidvid) = <collection logical id> “::” <collection version id>

specific identifier = <mission abbrev.> “_” <inst abbrev.> “_” <collection type>
[“_” <processing level or collection subtype>]
[“_” <target or target system or target type>]
[“_” <description>]

Mission abbreviation examples: (May also be the abbreviation for a spacecraft, telescope, observing campaign, or investigation)
CLEM
GLL
MGS
SL9

Instrument abbreviation examples:
MOC

NIMS
UVVIS

Collection types:
BROWSE
CALIBRATION
CONTEXT (for context products delivered as part of non-context bundles)
DATA
DOCUMENT
GEOMETRY (for non-SPICE geometry data)
INSTRUMENT
INSTRUMENT_HOST / OBSERVING_SYSTEM
MISSION / INVESTIGATION / CAMPAIGN
NODE
PUBLICATIONS
SPICE_KERNEL (for SPICE products delivered as part of non-SPICE bundles)
SUPPLEMENTARY
TARGET
XML_SCHEMA

Rules:

- char set =

7.1.4 Bundle Identifiers

bundle logical identifier (lid) = "URN:NASA:PDS:" <specific identifier>
bundle versioned identifier (lidvid) = <bundle logical identifier> "::" <bundle version id>

specific identifier = that portion of the collection specific identifier which is common across all the component collections of the bundle

bundle and collection ids must be unique across pds

in general they will be constructed major to minor order

the bundle id will consist of all those terms that are common across component collections

order of discriminating terms is up to data provider in consultation with node

underscores used within terms, dashes used between terms and = underscore

no forward slashes

LIDs are all uppercase (inside the labels; not enforced if used on file systems)

use ids rather than names for component terms (like “MGS” rather than “MARS GLOBAL SURVEYOR”)

7.2 Other Identifying Information

7.2.1 Version IDs

Version IDs are used for all types of products, including standard products, collections, and bundles.

Note that in all cases, the incrementing of a version identifier implies that a user of that product should use the latest version of the product. In other words, incrementing a version indicates that previous versions of the product have been superseded. (It is, of course, possible that a user may, in specific cases, deliberately choose to use older versions of a product. As long as those products were archived with the PDS, they will still be available, albeit potentially offline.)

Formation Rules:

- Version identifiers are appended to logical identifiers to form versioned identifiers. They are separated from the logical identifier by a double colon (“::”).

- Version IDs must be of the form M.m where “M” and “m” are both integers. “M” is the “major” component of the version and “m” is the “minor” component of the version.

- The major number is initialized to one for archive products. (Zero may be used for sample products or test run products that are not intended for the archive.) The minor number is initialized to zero.

- Whenever the major number is incremented, the minor number is re-set to zero.

- The minor portion of the version is **not** pre-padded with zeros; it is simply incremented as an integer. Thus, “1.1” and “1.10” are different versions; “1.01” is invalid.

It is up to the determination of each PDS discipline node, working in consultation with data preparers, to determine the criteria for how and under what circumstances version numbers will be incremented for a particular archive. Specific versioning conventions for every archive must be detailed in the respective product documentation. Note that depending upon the criteria that a given node/mission combination selects, it is possible in some cases that higher versioned products may be created without their lower versioned counterparts existing.

7.2.2 Local Identifiers

Change to ”Local IDs”?

Local identifiers are only required to be unique within an individual product label. They are primarily used for navigating among the different portions of the label and can be used, for example, to tie together a data object with an Object_Statistics class describing it, to to make clear the relationship between two Image_Grayscale classes (a primary and a browse) and the one or two File classes identifying the data they describe.

The construction of these identifiers is therefore largely up to the data provider, with the exception of the character set restrictions. There is also one recommendation, to make these identifiers simpler for humans to find in a label:

Rules:

- Must be unique within the containing label.
- The character set for local identifiers is restricted to ASCII letters, number, the underscore, the colon, the dash, and the space. Numeric codes for these ASCII characters to be added. Period? Forward slash? Semi-colon? - No space

formation rule: use class name with incrementing number appended, underscore, prepended with zeroes so that all local identifiers of the same class type have the same number of digits (This is to preserve the ability to do alphabetical sorting.)

7.2.3 Digital Object Identifiers

TBD – Probably need to add a description of these. – EDR

7.2.4 Uniform Resource Identifiers

TBD – Probably need to add a description of these. – EDR

7.2.5 Names

TBD - Is this still needed?

Rules:

-

Recommendations:

-

7.2.6 Titles and Alternate Titles

Titles and alternate titles are intended for human consumption. Their primary use is to be displayed alongside a product when it is shown in some sort of product browser. It is not required to be unique. The specificity of the title is entirely up to the data provider, and is constrained only the manpower and resources necessary to populate it. Both of the following are acceptable for titles:

"LORRI image of Io; lat: 4 degS, lon: 164 degW; time: 2007-03-01T00:35 UTC"

"This image of Io, captured by New Horizon's Long Range Reconnaissance Imager, captured the active Tvashtar plume. The image was acquired on March 1, 2007."

The first of these two titles could be generated automatically by data production pipeline software. The second required post-production modification to incorporate observed feature information. Either is acceptable.

UTF-8 printable characters (explicitly disallowing roughly 30 ASCII characters non-printable characters)

limit to 255 **bytes** in length (up to data providers to be careful if using UTF-8 to verify byte length) (character type definition?)

Rules:

- Titles must be no more than 255 characters in length.
- Titles are text strings. The character set is restricted to the US-ASCII? character set.

Recommendations:

- None.

How are alternate titles used?

7.3 Directories

Reserved directory names: browse, calibration, context, data, document, geometry, supplemental (or about), xml_schema

same character set as filenames except no periods

Rules:

-

Recommendations:

-

Although NTFS allows each path component (directory or filename) to be 255 characters long and paths up to about 32767 characters long, the Windows kernel only supports paths up to 259 characters long if no UNC is used for addressing. – Need to follow up on this. See the section on "Maximum Path Length Limitation" on the following web page: http://msdn.microsoft.com/en-us/library/aa365247(VS.85).aspx

7.4 Filenames

Although modern operating systems are extremely permissive when it comes to filenames, best practices argue against taking full advantage of this. For example, many characters that are legal in filenames require escaping and can create numerous problems when moving files from one operating system to another. Similarly, 255 character filenames, while permitted, are often inconvenient to users who can't distinguish between similarly named lengthy filenames in many views on various operating systems.

Reserved names: AUX, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, CON, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, NUL, PRN, (and none of the former with any extension), a.out, core, .profile, .history, and .cshrc

Rules:

- Filenames must be unique within directories.
- Filenames must be no longer than 255 characters (although see path length restrictions discussed in the section on directory names).
- Filenames must be case-insensitive. (In other words, "MyFile.txt" and "myfile.txt" are not permitted in the same directory.)
- The character set is restricted to A-Z (ASCII 0x41 through 0x5A), a-z (ASCII 0x61 through 0x7A), 0-9 (ASCII 0x30 through 0x39), dash "-" (ASCII 0x2D), underscore "_" (ASCII 0x5F), and period "." (ASCII 0x2E). (we may re-consider this later)
- Filenames may not begin or end with a dash, underscore, or period.
- Filenames must begin with an alphabetical character (A-Z or a-z or 0-9).

Recommendations:

- Don't abuse the permitted 255 character length for filenames. Try to keep them no longer than 30 to 40 characters in general. **Archives can fail peer review for being unnecessarily difficult to use as much as for violating actual PDS standards!**

Still have to discuss file extensions.

filenames must have at least one period followed by an extension. They may have more than one period, but PDS will consider all periods other than the final one to be part of the basename. (file base + file extension)

remember all SPICE file extensions need to be reserved

Need to include version information in filenames.

See: http://msdn.microsoft.com/en-us/library/aa365247(VS.85).aspx ("Naming Files, Paths, and Namespaces" from MSDN) and http://en.wikipedia.org/wiki/File_name (Wikipedia topic "Filename")

7.5 Classes

The following rules and recommendations for the naming of classes in the PDS data model apply to all levels of the model: common, node, and mission.

Rules:

- The character set is restricted to ASCII letters, numbers, and the underscore. The first character of the class name must be a letter.
- Each component of the class name shall begin with an uppercase character; all other characters shall be lowercase, except when the class name incorporates an acronym. (Ex. Stream_Delimited_Field, SPICE_Kernel, Array_2D)
- The class name must not exceed 255 characters.
- The word order for the components of the class name shall be most significant first (Ex. Product_Table_Character, rather than Character_Table_Product).

Recommendations:

- Whenever possible, when creating subclasses of existing classes, use portions of the parent class name in the child class name to make the relationship between the classes apparent. (Ex. Table_Base, Table_Base_Character or Image_3D, Image_Color, Image_Color_Anaglyph)
- Use widely recognized and accepted terms for class name components, that clearly indicate the nature of a particular class. (Examples to avoid: Image_PanMos_Proj_JMR, Fpu_Bin_Table, Comp_12_8_Parm)

Camel Case - MSB (or do what make sense) follows rules for sub-classes

7.6 Attributes

Rules:

-

Recommendations:

-

rightmost component must correspond to an existing data element concept no articles

tool that provides pull down list of data element concepts for user to select from,

lower case - LSB

rules for common and node dictionaries but not for mission?

7.7 Attribute Values

Rules:

-

enumerated values - case no restrictions, but must match enumerated values (i.e. no limitation on person who created dictionary, but limitation on data preparer is that they must follow DD), UTF-8

Recommendations:

-

Part II

Data Content Standards

Chapter 8

Astronomical Nomenclature

Naming conventions for astronomical objects have long been under the purview of the IAU and its assorted Working Groups. Data archived with the PDS shall utilize the naming conventions for both astronomical bodies and features established by the IAU. The following information is taken directly from the IAU web site. In all cases of conflict or contradictions between the information provided in this chapter and the IAU, the conventions and rulings of the IAU shall take precedence.

http://www.iau.org/public/naming/

According to the posted IAU copyright information (http://www.iau.org/copyright), the information posted on their web site is copyright-free and may be reproduced without fee, as long as we credit them. They do stipulate, however, that a copy of the "product" (this document, presumably) be sent to them. Are we content with this, or should I re-write the following material? - EDR

8.1 Spelling of Names

Questions have been asked about the proper English spelling of names of astronomical objects, especially as regards capitalization of such names.

The IAU formally recommends that the initial letters of the names of individual astronomical objects should be printed as capitals (see the IAU Style Manual, Trans. Int. Astron. Union, volume 20B, 1989; Chapter 8, page S30 – PDF file; e.g., Earth, Sun, Moon, etc. "The Earth's equator" and "Earth is a planet in the Solar System" are examples of correct spelling according to these rules.

It is emphasized, however, that language conventions are the responsibility of individual nations or groups of nations. While the IAU is willing to help to achieve a minimum degree of orthographic consistency as regards astronomical terms, it cannot undertake to do so for all languages, nor is it in the power of the IAU to enforce the application of any such conventions.

8.2 Naming of Planets and Planetary Features in the Solar System

When the first images of the surface of a planet or satellite are obtained, themes for naming features are chosen and names of a few important features are proposed, usually by members of the appropriate IAU task group. Later, as higher resolution images and maps become available, names for additional features may be requested by investigators mapping or describing specific surfaces or geological formations. Anyone may suggest that a specific name be considered by a task group, but there is no guarantee that the name will be accepted. Names successfully reviewed by a task group are submitted by the task group chair to the Working Group for Planetary System Nomenclature (WGPSN). Upon successful review by the members of the WGPSN, names are considered approved and can be used on maps and in publications. Approved names are immediately entered into the Gazetteer of Planetary Nomenclature, and posted on its web site. Any objections to these names based on significant, substantive problems must be forwarded in writing or email to the IAU Division III President within three months from the time the name was placed on the web site. Approved names are also listed in the transactions of the IAU.

(from http://planetarynames.wr.usgs.gov/approved.html)

More information on planetary nomenclature can be found on the USGS FAQ section
http://planetarynames.wr.usgs.gov/nomenFAQ.html

8.2.1 Definition of a Planet

We invite you to consult the IAU Resolutions 5 and 6 (PDF file, 92KB) adopted on August 2006, at our XXVIth General Assembly in Prague.

8.3 Satellites of Planets in the Solar System

The number of planetary satellites now (April 2008) stands at 165. The CCD technology has made it possible to discover satellites down to 1 km in size. At some time in the future it may be advisable to stop naming very small satellites. The greatly increased discovery rate of satellites has made it necessary to extend the existing name categories for the satellites of Jupiter and Saturn whose names are drawn from the Greco-Roman mythology. The Jovian satellites have previously been named for Zeus/Jupiter's lovers and favorites but now Zeus' descendants are also included as an allowable source of names. The satellites of Saturn have so far been named for the Greco-Roman Titans, descendants of the Titans, Giants and the Roman god of the beginning. In order to internationalize the names, we now also allow names of giants and monsters in other mythologies (so far Gallic, Inuit and Norse).

More information

http://planetarynames.wr.usgs.gov/append7.html

The latest updates on nomenclature of features on the surface of satellites (craters, mountains, valleys, etc) can be found on the Planetary Surface Feature News page.

8.4 Minor Planets

The discoverer of a particular object has the privilege of suggesting a name to a special Committee of the IAU that judges its suitability. Contrary to some recent media reports it is not possible to buy a minor planet. If you have a name you would like to apply to a minor planet, the best advice is "Go out and discover one!".

8.4.1 A more detailed description of how Minor Planets are named:

The assignment of a particular name to a particular minor planet is the end of a long process that can take many decades. It begins with the discovery of a Minor Planet that cannot be identified with any already-known object. Such Minor Planets are given a provisional designation. More detailed info about the whole process from discovery to numbering of newly discovered Minor Planets can be found here or here. The provisional designations are based on the date of discovery and are assigned by the Minor Planet Center MPC according to a well defined formula that involves the year of discovery, two letters and, if need be, further digits (for example 1989 AC or 2002 LM60).

When the orbit of a Minor Planet becomes well enough determined that the position can be reliably predicted far into the future (typically this means after the Minor Planet has been observed at four or more oppositions), the Minor Planet receives a permanent designation - number issued sequentially by the Minor Planet Center, for example (433), (4179) or (50000).

When a Minor Planet receives a permanent number, the discoverer of the Minor Planet is invited to suggest a name for it. The discoverer has this privilege for a period of ten years following the numbering of the object.

The discoverer writes a short citation explaining the reasons for assigning the name according to the guidelines of the IAU.

All proposed names are judged by the fifteen-person Working Group for Small Body Nomenclature (CSBN) of the IAU, comprised of professional astronomers with research interests connected with Minor Planets and/or comets from around the world.

Proposed names should be:

- 16 characters or less in length
- preferably one word
- pronounceable (in some language)
- non-offensive
- not too similar to an existing name of a Minor Planet or natural Planetary satellite.

The names of individuals or events principally known for political or military activities are unsuitable until 100 years after the death of the individual or the occurrence of the event.

In addition,

- names of pet animals are discouraged
- names of a purely or principally commercial nature are not allowed.

There are more detailed guidelines for unusual Minor Planets in certain dynamical groups, for example:

- Trojan asteroids (those that librate in 1:1 resonance with Jupiter) are named for heroes of the Trojan War (Greeks at L4 and Trojans at L5).
- Trans-Jovian Planets crossing or approaching the orbit of a giant Planet but not in a stabilizing resonance (so called Centaurs) are named for centaurs.
- Objects crossing or approaching the orbit of Neptune and in stabilizing resonances other than 1:1 (notably the Plutinos at the 2:3 resonance) are given mythological names associated with the underworld

- Objects sufficiently outside Neptune's orbit that orbital stability is reasonably assured for a substantial fraction of the lifetime of the solar system (so called Cubewanos or ”classical” TNOs) are given mythological names associated with creation.

- Objects that approach or cross Earth's orbit (so called Near Earth Asteroids) are generally given mythological names.

Accepted names become official when they are published, along with their accompanying citations, in the Minor Planet Circulars, issued monthly by the Minor Planet Center.

The Committee on Small Body Nomenclature (CSBN) recognizes the need to limit the numbers of Minor Planets named, and it requests individual discoverers and teams to propose no more than two names each two months.

The alphabetic list of all names is available at the Minor Planet Center including the discovery circumstances.

Chapter 9

Calibration Standards

TBD - Will not be completed for first release.

Chapter 10

Cartographic Standards

Pulled fairly verbatim from v3.8 of StdRef. Needs to be adapted to PDS4.

10.1 Introduction

To facilitate use, exchange and integration of its products, the PDS follows accepted planetary cartographic standards for data products where they exist. Because such standards evolve as new data and knowledge are acquired, there are advisory groups charged with developing and periodically updating standards for coordinate systems. All data providers for PDS products should follow accepted standards and be aware of current NASA and international recommendations on cartographic coordinate systems and conventions relevant to their bodies of interest. An absolute requirement for all PDS products is that relevant coordinate systems and frames be clearly specified in product labels and supporting documents. This chapter specifies, as of late 2008, the authoritative sources for international cartographic standards, provides a summary of major cartographic elements to which those standards apply, and identifies the primary standards that PDS has adopted.

10.1.1 International and NASA Advisory Groups for Cartographic Standards

The primary international body for coordinate systems in the Solar System is the International Astronomical Union (IAU). The IAU has recognized the International Celestial Reference System (ICRS) as the defining inertial reference system and its associated International Celestial Reference

Frame (ICRF) (Ma et al., 1998) as the defining frame for that system. The ICRS and ICRF are maintained for the IAU by the International Earth Rotation and Reference Systems Service (IERS, http://www.iers.org/).

For cartographic coordinates and conventions for planets and satellites, the IAU and the International Association of Geodesy (IAG) have established jointly the Working Group on Cartographic Coordinates and Rotational Elements (WGCCRE), which publishes triennial reports, currently in the journal Celestial Mechanics and Dynamical Astronomy (Davies, et al., 1980, 1983, 1986, 1989, 1992, 1996; Seidelmann, et al., 2002, 2005, 2007). This working group includes PDS-affiliated scientists, thus assuring full interaction in defining the standards. Publications and reports issued by the WGCCRE can be found at http://astrogeology.usgs.gov/Projects/WGCCRE/. PDS data providers should refer to these reports for current information and recommendations on rotational elements for Solar System bodies and how these are related to their cartographic coordinates.

The NASA Lunar Geodesy and Cartography Working Group and the Mars Geodesy and Cartography Working Group are sponsored by the NASA Lunar Precursor Robotics Program (LPRP) and Mars Program offices, respectively, and are responsible within NASA for providing additional coordination of cartographic standards and related (e.g., data processing) issues (Archinal et al., 2008a, 2008b; Duxbury et al., 2002). These Working Groups have made additional recommendations regarding coordinate systems (generally with additional detail) beyond those of the WGCCRE.

10.2 Inertial Reference Frame and Time System

The orientation of a body in the Solar System can be calculated using a series of rotation angles to define the directions of the bodys principal axes with respect to an inertial reference frame (i.e., a system that is not rotating or accelerating relative to a specific reference point) which provides a standard frame from which position, velocity, and acceleration can be measured. Such a reference frame is a set of identifiable fiducial points and their positions on the sky, providing a practical realization of a reference system that defines the origin, fundamental planes (or axes), and transformations between observed elements and reference points in the celestial coordinate system. Reference coordinate systems are defined by a system of concepts (e.g., using planetocentric latitude and longitude) while a reference coordinate frame is a specific realization of a coordinate system that is anchored to real data (such as a photogrammetric control network, altimetry crossover solutions, or lunar ephemerides) (Kovalevsky and Mueller, 1981).

For a planetary body in space, position is defined relative to a Z axis (typically the spin vector of the body, or the planetographic north pole), the X axis (defined as the point where the equator of the body crosses the equatorial plane of an inertial frame at a specific epoch), and the Y axis of a right-handed system. The standard units for coordinates are based on the International System of

Units (SI), including decimal degrees. The orientation of Solar System bodies can be calculated from angular position (right ascension ? and declination ??) with respect to the equatorial system of a particular epoch. For example, the orientation of the north pole of a body at a given epoch is specified by its right ascension ? and declination ?, while the location of the prime meridian is specified by the angle W (Davies et al., 1980).

The standard epoch is called J2000.0 and is defined to be 2000 January 1.5 TDB, where TDB is Barycentric Dynamical Time (e.g., Seidelmann et al., 2007). This corresponds to 2000 January 1, 1200 hours TT (Terrestrial Time) or the Julian Date 2451545.0 (NAO, USNO and HMNAO, 1983). This also corresponds to 2000 January 1, 11:58:55.816 UTC (Coordinated Universal Time; Seidelmann et al., 1992). Although the natural system for many applications would be TDB, UTC is considered the fundamental system for all PDS data products. The standard way of expressing UTC is in year, month, day, hour, minute, and decimal seconds. Julian Dates (JD) are supported as a supplementary system for reporting UTC time. However the JD time scale must be specified (e.g., UTC or TDB). See the Planetary Science Data Dictionary (PDS, 2008), chapter 2, for further information on time representation.

The currently accepted orientation of the inertial system (i.e., J2000.0 right ascension and declination) is defined by the International Celestial Reference System (ICRS), which is a particular implementation of the Barycentric Celestial Reference System (BCRS) (IAU, 2000). The ICRS is the fundamental celestial reference system of the IAU, and it has an origin at the barycenter of the Solar System and space fixed (kinematically non-rotating) axis directions. As noted by the IAU, the ICRS is meant to represent the most appropriate coordinate system for expressing reference data on the positions and motions of celestial objects. Specifications for the ICRS include a metric tensor, a prescribed method for establishing and maintaining axis directions, a list of benchmark objects with precise coordinates, and standard algorithms to transform these coordinates into observable quantities for any location and time. The ICRS is derived from the International Celestial Reference Frame (ICRF) comprised of coordinates for a set of fiducial points on the sky. The ICRF is within 0.05 arcseconds (Chapront et al., 2002; Herring et al., 2002) of the Solar System inertial frame based on Earths Mean Equator (EME) at the Equinox of Julian Ephemeris Date (JD) 2451545.0 (i.e., J2000.0). This is consistent with current dynamical practice and spacecraft and planetary ephemerides (e.g., those provided by the NASA Jet Propulsion Laboratory).

Many older data sets, collected before the J2000.0 system and ICRF were defined, are referenced to EME and Equinox of Besselian 1950.0 (B1950.0; JD 2433282.423). While this reference frame should not be used for current data, PDS supports this reference frame for older data. Transformation between the B1950.0 and J2000.0 (and the nearly equivalent ICRF) systems has been well defined by the IAU (NAO, USNO and HMSNAO, 1983; also see http://nedwww.ipac.caltech.edu/forms/calculato

Positions may be expressed in other coordinate systems and associated frames, which can be derived from the fundamental system and frame, when this enhances the use of the data for various applications. These include ecliptic-based coordinates and heliographic coordinates. These coordinates, while possibly ”natural” for many applications, are derivable from the fundamental system and are therefore treated as supplementary data by PDS. In some cases, it is convenient to work in one preferred coordinate system and then to convert to another, more standard system for products. This practice of providing the natural working coordinates in addition to the coordinates in a fundamental system promotes ease of use of PDS products and should be adopted by all data providers who use coordinate systems other than the fundamental system. As noted above, all supplementary coordinate systems must be fully documented in PDS products and must be negotiated with the PDS prior to delivery.

10.3 Spin Axes and Prime Meridians

The spin axis orientations of many Solar System bodies are defined by the WGCCRE in the ICRF inertial reference frame. For historical reasons, the orientation of the spin axis of planets and satellites is defined by the north pole, which is the pole that is on the northern side of the Invariant Plane of the Solar System (close to but not the same as the ecliptic). With this definition of the north pole, it is also necessary to specify whether the rotation is direct or prograde (in the same direction as the Suns rotation or counterclockwise when viewed from above the north pole) or retrograde (opposite to the direction of the Suns rotation).

For small bodies such as comets and asteroids, for which precession due to torques can cause large changes in the angular momentum vector, the orientation is defined by the positive pole, which is the pole determined by the right hand rule for rotation. Since some small bodies can be in excited state rotation, there are numerous complications in application that are addressed in more detail in the WGCCRE reports. Depending on the mode of excited state rotation, the axis may coincide with the maximum moment of inertia. Some cases, particularly the case of chaotic rotation, are considered on a case by case basis by the WGCCRE.

If a body has a solid surface, prime meridians for a given longitude system may be defined by specifying the coordinates of a surface feature on the body (usually a small feature such as a crater in the equatorial region) or by the mean direction relative to the parent body for synchronously rotating bodies (e.g., the Moon, the Galilean moons, and most of the Saturnian moons). Where insufficient observations exist to determine the principal moment of inertia, coordinates of a surface feature will be specified and used to define the prime meridian. In the case of planets without solid surfaces, the definition of the prime meridian is somewhat arbitrary. In any case, the actual definitions are decided by the WGCCRE, not by the PDS. We note that influxes of new data often lead to an iterative process to define (or improve) the orientation of the spin axis or other parameters used to define a coordinate system and in these cases the data providers (e.g., spacecraft mission personnel) and the WGCCRE must maintain close contact regarding the definition.

10.4 Body-Fixed Planetary Coordinate Systems

Two types of coordinate systems are fixed to the body planetocentric and planetographic. Details of the coordinate systems for planets and satellites differ from those for small bodies and rings. This section discusses only the aspects that are common to all applications. The Planetocentric system has an origin at the center of mass of the body. Planetocentric coordinates are defined by a vector from the center of mass of the body (often approximated as the center of figure) to the point of interest, typically but not necessarily a point on the surface (e.g., an impact crater with known position). The planetocentric latitude is the angle between the equatorial plane and the vector, while the planetocentric longitude is the angle between the prime meridian and the projection of the vector onto the equatorial plane.

The Planetographic system also has an origin at the center of mass of the body. Planetographic coordinates, however, are defined by vectors perpendicular to a reference surface, often a biaxial ellipsoid that is centered on the body and chosen to describe the gross shape of the body. Reference surfaces vary from body to body and are defined by the WGCCRE in consultation with the observers who provide the information to define such surfaces. The most common reference surface is an oblate spheroid aligned with the spin axis of the body. However, for certain applications the reference surface may be a triaxial ellipsoid, a gravitational equipotential, or a higher order surface model.

For a biaxial ellipsoid the planetographic latitude is the angle between the equatorial plane and a vector through the point of interest, where the vector is normal to the reference surface. Planetographic longitude is the angle between the prime meridian and the projection of the same vector onto the equatorial plane. In general, the planetographic vector does not pass through the origin. The vector need not pass through the spin axis but in most realistic cases it does. If the reference surface is a sphere, the planetographic and planetocentric vectors are identical.

The WGCCRE allows for the use of either planetographic or planetocentric coordinates for a given body, so data providers may adopt either system. Historically planetographic coordinates have been preferred for cartographic products, while planetocentric coordinates were used for dynamical (i.e. orbit, gravity field, altimetric) observations and calculations. For the planet Mercury, the MESSENGER mission has chosen to use planetocentric coordinates as the primary coordinate system for all products (Seidelmann et al., 2007). For the planet Mars, the MGCWG and all current NASA missions have chosen to use planetocentric coordinates as the primary coordinate system for products (Duxbury et al., 2002). Producers of printed or electronically printed maps (e.g., in PDF format) may wish to show both types of coordinates.

10.4.1 Planets and Satellites

For planets and satellites, the conventions are complicated for historical reasons. In the planetocentric coordinate system, northern latitudes are those in the hemisphere of the body containing the spin pole that points to the northern side of the invariant plane of the Solar System. The bodys rotation direction, either prograde or retrograde, must also be specified. Planetocentric longitude increases eastward (i.e., in the direction defined by the right-hand rule and the north pole) from the prime meridian, from 0 to 360. Thus an external observer sees the longitude decreasing with time if the rotation is prograde but increasing with time if the rotation is retrograde.

North and south planetographic latitude are defined in the same way as for planetocentric latitude, although the numerical values for a given point on the surface, (other than on the equator or at the poles) are different if the reference surface is not a sphere. The definition of planetographic longitude is dependent upon the rotation direction of the body, with the basic definition being that an external observer should see the longitude increasing with time, or that the longitude increases in the direction opposite to the rotation, although there are exceptions due to historical practice for Earth, the Moon, and Sun. That is to say, the longitude increases to the west if the rotation is prograde (or eastward) and vice versa. Whether the rotation direction is prograde or retrograde can be determined from the current WGCCRE report. See Tables 1 and 2 (or their equivalent in any future report), where the sign of the velocity term for W indicates either prograde (positive) or retrograde (negative) rotation. For all bodies a longitude range of 0 to 360 can be used.

For Earth, the Moon, and the Sun, a longitude range of -180 to +180 has been used in the past [including in existing PDS data sets, as defined by the Planetary Science Data Dictionary (PDS, 2002)] and is allowed by the WGCCRE. However, for the Moon, the NASA LGCWG and LRO Mission recommend that in the future, only the 0 to 360 range be used (LGCWG, 2008; LRO Project, 2008). For printed or electronically printed maps (e.g., in PDF format), it may be useful to label the longitude grid both with primary 0 to 360 coordinates and -180 to +180 coordinates.

For the Moon, two slightly different reference systems are commonly used to orient the lunar body-fixed coordinate system. One is the Mean Earth/Polar Axis (ME) system, the preferred system to be used for PDS data products. The other is the axis of figure system, also called the Principal Axis (PA) system, sometimes used internally among instrument teams for specific applications. For computing precise lunar coordinates, the WGCCRE recommends the use of the JPL DE403 ephemeris (which provides lunar orientation in the PA system), rotated into the ME system. The WGCCRE noted in its most recent report that improved versions of the JPL ephemerides were imminent and might be used instead. In fact the JPL DE421 ephemeris is now available and, after rotation into the ME system, is recommended for use (LGCWG, 2008; LRO Project, 2008). The maximum difference between these two frames in the ME system for the period 2000-2019 is only about 6 meters (Archinal, 2008).

10.4.2 Small Bodies

For small bodies (asteroids and comets), both planetographic and planetocentric coordinates follow the same right hand rule that is used to define the positive pole, which can be either above or below the invariant plane of the Solar System. For the simple case of a body with positive pole pointing to the northern hemisphere of the Solar System, this corresponds to longitude, both planetocentric and planetographic, increasing eastward, 0 to 360, which in turn corresponds to the case in which the longitude seen by an outside observer decreases with time.

For some small bodies, coordinates based on latitude and longitude alone can be multi-valued in radius i.e., the vector from the center of the body can intersect the surface in more than one place. There may also be complications (due to the irregular shape) which force special procedures when producing a useful, planar map. Such details are discussed in reports of the WGCCRE.

10.4.3 Rings

There is no international standard for ring coordinate systems. Standards in use for such PDS products were defined by experts in the Rings Node, in consultation with a broad cross-section of interested scientists. Conventions for coordinate systems for rings are similar to those for small bodies, in as much as they are all based on a right-hand rule, with longitude increasing in the direction of orbital motion. Thus longitude increases eastward for the prograde-moving rings (Jupiter, Saturn, and Neptune), but it increases westward for retrograde-moving rings of Uranus. Rings also use a positive pole direction following the right hand rule, analogous to the case for small-body rotation, thus coinciding with the North Pole of Jupiter, Saturn, and Neptune, but the South Pole of Uranus.

Coordinates for rings differ from those for planets and small bodies in not being body-fixed because there are no fixed features to define longitude. They are defined in an inertial system that is co-moving with the center of mass of the parent body. Specifically, longitudes are measured from the ascending node of the plane of the rings in the ICRF, i.e. the point at which the plane of the rings intersects the ICRF equator. In the case of inclined rings, longitudes are measured as a broken angle from the ascending node of the planets equatorial plane in the ICRF, along the equatorial plane to the ring planes ascending node, and thereafter along the ring plane.

10.4.4 Planetary Plasma Interactions

There are no international standards for values or names of coordinate systems of planetary plasma observations. Recommendations for coordinate systems in the near-Earth environment by Russell

(1971) have been generalized for use with plasma observations at other bodies. More recently, other systems have been defined (e.g., Franz and Harper, 2002) and are currently in use. The coordinate systems used for plasma observations and data analysis typically are right-handed. The primary exception to this rule is the left-handed Jovian System III.

Standards for planetary plasma data products for PDS were defined by experts in the Planetary Plasma Interactions Node, following recommendations from Russell (1971) and Franz and Harper (2002) and in consultation with other specialists. Providers and users of PDS data featuring plasma observations are encouraged to use names as defined by these authors where appropriate, and to follow similar name construction when new systems must be defined.

10.5 Surface Models

A standard reference surface model commonly used for hard surfaces is the digital terrain model (DTM). The DTM defines body radius or geometric height above the body reference surface as a function of cartographic latitude and longitude. Spheroids, ellipsoids and harmonic expansions giving analytic expressions for radius as a function of cartographic coordinates are all allowed in PDS. A DTM may also define potential height, i.e., elevation, above an equipotential surface, provided the method is specified, including the specification of appropriate constants and gravity field that is used to convert to/from radii and potential height.

The only internationally recognized DTM is the MOLA model for Mars (Seidelmann, et al., 2007, page 168 in WGCCRE #10). DTMs are also available for other bodies, including the Moon and several small bodies; but their use is not officially recommended and therefore up to the individual user.

The digital image model (DIM) defines body brightness in a specified spectral band or bands as a function of cartographic latitude and longitude. A DIM may be associated with the surface radius, geometric height, or potential height values in a corresponding DTM or it may be registered independently to a spheroid, ellipsoid, or spherical harmonic expansion.

10.6 PDS Keywords for Cartographic Coordinates

NOTE: This section definitely needs updating for PDS4!!!

To support the descriptions of these various reference coordinate systems and frames, the PDS has defined the following set of geometry data elements [see the Planetary Science Data Dictionary (PDS, 2008) for complete definitions and additional data elements].

A_AXIS_RADIUS
B_AXIS_RADIUS
C_AXIS_RADIUS
COORDINATE_SYSTEM_CENTER_NAME
COORDINATE_SYSTEM_DESC
COORDINATE_SYSTEM_ID
COORDINATE_SYSTEM_NAME
COORDINATE_SYSTEM_REF_EPOCH
COORDINATE_SYSTEM_TYPE
EASTERNMOST_LONGITUDE
LATITUDE
LONGITUDE
MAXIMUM_LATITUDE
MAXIMUM_LONGITUDE
MINIMUM_LATITUDE
MINIMUM_LONGITUDE
POSITIVE_LONGITUDE_DIRECTION
WESTERNMOST_LONGITUDE

To support the description of locations in a planetary ring system, the PDS has defined the following data elements:

CENTER_RING_RADIUS
RING_RADIUS
MINIMUM_RING_RADIUS
MAXIMUM_RING_RADIUS

RING_LONGITUDE
MINIMUM_RING_LONGITUDE
MAXIMUM_RING_LONGITUDE

B1950_RING_LONGITUDE
MINIMUM_B1950_RING_LONGITUDE
MAXIMUM_B1950_RING_LONGITUDE

RING_EVENT_TIME
RING_EVENT_START_TIME
RING_EVENT_STOP_TIME

RADIAL_RESOLUTION
MINIMUM_RADIAL_RESOLUTION
MAXIMUM_RADIAL_RESOLUTION

The radius and longitude elements define an inertial location in the rings, and the ring event time elements define the time at the ring plane to which an observation refers. If desired, the radial resolution elements can be used to specify the radial dimensions of ring features that can be resolved in the data. See the Planetary Science Data Dictionary (PSDD; PDS, 2008) for complete definitions of these elements.

Some rings are not circular and/or equatorial. In these cases, the PSDD provides additional elements that can be used to describe a rings shape. The elements are:

RING_SEMIMAJOR_AXIS
RING_ECCENTRICITY
RING_PERICENTER_LONGITUDE
PERICENTER_PRECESSION_RATE
RING_INCLINATION
RING_ASCENDING_NODE_LONGITUDE
NODAL_REGRESSION_RATE
REFERENCE_TIME

Here the value of REFERENCE_TIME indicates the instant at which the LONGITUDE elements are defined. The actual pericenter and ascending node at the time of an observation are determined based on the precession and regression rates as follows:

```
pericenter_longitude = RING_PERICENTER_LONGITUDE +
        PERICENTER_PRECESSION_RATE *
        (observation_time - REFERENCE_TIME) mod 360
  ascending_node_longitude =
        RING_ASCENDING_NODE_LONGITUDE +
        NODAL_REGRESSION_RATE *
        (observation_time - REFERENCE_TIME) mod 360
```

The oscillating modes of a ring can also be specified if necessary:

RING_RADIAL_MODE
RING_RADIAL_MODE_AMPLITUDE
RING_RADIAL_MODE_FREQUENCY
RING_RADIAL_MODE_PHASE

Additional elements should be used to specify the assumed orientation of the planets pole:

POLE_RIGHT_ASCENSION
POLE_DECLINATION
COORDINATE_SYSTEM_ID

The COORDINATE_SYSTEM_ID can be either J2000.0 or B1950.0, with J2000.0 serving as the default. See the PSDD for further details.

10.7 Map Resolution

A uniform set of resolutions is helpful for analyses of multiple datasets and development of map products derived from PDS data, and the selected scale must account for differences in available image resolution and quality. Such map scales are measured against a reference surface that is typically a geometrically defined shape that represents a given planetary body. For global maps, the recommended spatial resolution for a map is 2n pixels per degree of latitude, where a pixel is treated as a finite area and n is an integer. A spatial resolution of 2n pixels per degree allows simple coregistration of multiple datasets by doubling or halving the pixel sizes (typically by averaging or interpolation) and without resampling or otherwise changing the pixels. These recommendations continue a convention established in the 1960s and 1970s by the lunar and Mars research communities (e.g., Batson, 1987; Greeley and Batson, 1990), as advocated by the NASA Planetary Cartography Working Group (PCWG) and its successor the Planetary Cartography and Geologic Mapping Working Group (PCGMWG) (PCWG, 1993, pp. 22-24), and affirmed by the LGCWG (2008).

For polar regions of global maps, the recommendation is also to use the binary map scale or 2n pixels per degree of latitude near the pole. This practice maintains consistency with the global data product.

For working at landing site scales with data that has pixels of tens of centimeters to a few meters in size, spatial resolutions of maps are more convenient if provided at scales of 1 meter per pixel resolution or multiples thereof (LGCWG, 2008). At such human scales this convention is simpler and will preserve inherent details of resolution for applications such as landing site operations, traversing, and surface engineering studies.

For both global and local maps showing elevation or relief, the recommended vertical resolution is 1 x 10m meters, where m is an integer chosen to preserve all the resolution inherent in the data.

10.8 References

(Need to fix references.)

(Note: All WGCCRE reports are listed below for completeness. WGCCRE Report 7 was not issued).

Archinal, B. A. (2008). Summary of Lunar Geodesy and Cartography Working Group Teleconference of Tuesday, 2008 April 16, May 18.

Archinal, B. A. and the Lunar Geodesy and Cartography Working Group (2008a). Lunar Mapping Standards and the NASA LPRP Lunar Geodesy and Cartography Working Group, Scientific Event B01, The Moon: Science, New Results, Ongoing Missions, Future Robotic and Human Exploration,, 37th COSPAR Scientific Assembly, July 13-20, Montreal, Canada, http://www.cospar-assembly.org/home.php. Abstract B01-0050-08 and available at http://www.cospar-assembly.org/user/download.php?

Archinal, B. A. and the Lunar Geodesy and Cartography Working Group (2008b). Lunar Science Support Activities by the NASA LPRP Lunar Geodesy and Cartography Working Group: Recommendations for Lunar Cartographic Standards, NLSI Lunar Science Conference, July 20-23, Moffett Field, CA. Abstract no. 2080, available at http://www.lpi.usra.edu/meetings/nlsc2008/pdf/2080.pdf.

Chapter 11

Context Information

Still needs some rewriting.

Scientific data cannot be properly understood and interpreted without knowledge of the context in which the data was acquired. The PDS requires that contextual information for all archived data be included within the archive or be identified and readily available from external publishers.

This includes such varied information as the mission, spacecraft or other facility within which the instrument operates, the instrument that acquired the data, the target of the investigation, and an overview of the data collection. Additional documentation should be provided on the key personnel associated with the instrument, data, and archiving task. Finally, any external references providing additional, supplementary information about the data should also be included.

PDS maintains a set of context files online for use where relevant.

Much context information evolves during the course of a mission and necessitates update to the context documentation. For example, unexpected events which occur during the course of a mission will necessitate updates documenting the impact of various instrument and spacecraft systems or on the quality of the data. This might include such as 'safing' incidents, instrument anomalies or failures, extensions of the mission, etc.

11.1 Investigations

Investigation is a super-category including sub-classes of Mission, Observing Campaign, and Other Investigation. The information may be included directory in the archive or referenced in an external journal article. It must be provided and updated by the flight project or other coordinating authority.

11.2 Nodes

This information describes the PDS node responsible for curating the archived data. The PDS node is responsible for providing the information.

11.3 Observing Systems

11.3.1 Instrument Hosts

Instrument_Host is a generic term covering any kind of platform on which an instrument may be mounted. The PDS currently recognizes three types of platforms: Laboratory, Rover, and Spacecraft.

11.3.2 Instruments

This information may be either included directly in the PDS archive or located in a journal article published by an external organization. In either case, it is the responsibility of the instrument team or hosting observatory to provide the information.

11.4 Personnel

Indicates PDS, mission, and other personnel involved in the production of PDS archives. Includes two sub-categories: PDS Affiliate and PDS Guest. This information will be provided by the investigator.

11.5 Publications / References

TBD

11.6 Resources

TBD

11.7 Software

TBD

11.8 Targets

PDS now recognizes multiple targets for instrument observations including: calibration objects, dust, features, regions, rings, solar system bodies, solar wind, and stars. This information will be provided by the PDS, gleaned from external sources.

Chapter 12

Documentation

Supplementary or ancillary reference materials are usually included with archive products to improve their short- and long-term utility. These documents augment the internal documentation of the product labels and provide further assistance in understanding the data products and accompanying materials. Typical archive documents include:

- Flight project documents
- Instrument papers
- Science articles
- Volume information
- Software Interface Specifications (SISs)
- Software user manuals

The PDS criteria for inclusion of a document in the archive are:

1. Would this information be helpful to a data user?
2. Is the material necessary?
3. Is the documentation complete?

In general, the PDS seeks to err on the side of completeness.

Each document to be archived must be prepared and saved in a PDS-compliant format, including a PDS label. Documents are delivered in the DOCUMENT directory of an archive volume (see the Volume Organization and Naming chapter of this document).

A flat, human-readable ASCII text version of each document must be included on the volume, although additional versions may be included in other supported formats at the option of the data producer. Flat ASCII text means the file may contain only the standard, 7-bit printable ASCII character set, plus the blank character and the carriage-return and linefeed characters as record delimiters. A file is human-readable if it is not encoded and if any special markup tags which may be included do not significantly interfere with an average users ability to read the file. So, for example, simple HTML files and TeX/LaTeX files with relatively little markup embedded in the text are generally considered human-readable and may, therefore, be used to satisfy the above ASCII text version requirement.

Note that the PDS takes the requirement for complete documentation very seriously. Documents that are essential to the understanding of an archive are considered as important as the data files themselves. Furthermore, including a document in a PDS archive constitutes publication (or re-publication) of that document. Consequently, documents prepared for inclusion in an archive are expected to meet not only the PDS label and format requirements, but also the structural, grammatical and lexical requirements of a refereed journal submission. Documents submitted for archiving which contain spelling errors, poor grammar or illogical organization will be rejected and may ultimately lead to the rejection of the submitted data for lack of adequate documentation.

Chapter 13

Time Standards

TBD from ISO 8601

The following is a place holder and needs to be updated.

PDS has adopted a subset of the International Standards Organization Standard (ISO/DIS) 8601 standard entitled Data Element and Interchange Formats - Representations of Dates and Times, and applies the standard across all disciplines in order to give the system generality.

It is important to note that the ISO/DIS 8601 standard covers only ASCII representations of dates and times.

13.1 Date/Times

In the PDS there are two recognized date/time formats: Now incorrect!

CCYY-MM-DDTHH:MM:SS.sss (preferred format)
CCYY-DDDTHH:MM:SS.sss

Each format represents a concatenation of the conventional date and time expressions with the two parts separated by the letter T:

CC - century (00-99)
YY - year (00-99)
MM - month (01-12)
DD - day of month (01-31)

DDD - day of year (001-366)
T - date/time separator
HH - hour (00-23)
MM - minute (00-59)
SS - second (00-59)
sss - fractions of second (000-999)

Note: See Section 7.4 Midnight and Leap Seconds for special cases involving the indication of midnight and leap seconds.

The preferred date/time format is: CCYY-MM-DDTHH:MM:SS.sss.

Date/Time Precision

The above date/time formats may be truncated on the right to match the precision of the date/time value in any of the following forms:

1998
1998-12
1998-12-01
1998-12-01T23
1998-12-01T23:59
1998-12-01T23:59:58
1998-12-01T23:59:58.1
1998-12-01T23:59:58.12

13.2 Dates

Dates should be expressed in the conventional ISO/DIS 8601 format. On those rare occasions when dates cannot be expressed in the conventional format, a native format may be used.

13.2.1 Conventional Dates

Conventional dates are represented in ISO/DIS 8601 format as either year (including century), month, day-of-month (CCYY-MM-DD), or as year, day-of-year (CCYY-DDD). The hyphen character (-) is used as the field separator in this format. The year, month, day-of month format is the preferred format for use in PDS labels and catalog files and is referred to as PDS standard date format, but either format is acceptable.

13.2.2 Native Dates

Dates in any format other than the ISO/DIS 8601 format described above are considered to be in a format native to the specific data set, thus native dates. Native date formats are specified by the data preparer in conjunction with the PDS data engineer. Mission-elapsed days and time-to-encounter are both examples of native dates.

13.3 Times

The PDS allows times to be expressed in conventional and native (alternate) formats.

13.3.1 Conventional Times

Conventional times are represented as hours, minutes and seconds according to the ISO/DIS 8601 time format standard: HH:MM:SS[.sss]. Note that the hours, minutes, and integral seconds fields must contain two digits. The colon (':') is used as a field separator. Fractional seconds consisting of a decimal point (the European-style comma may not be used) and up to three digits (thousandths of a second) may be included if appropriate.

Coordinated Universal Time (UTC) is the PDS time standard and must be formatted in the previously described ISO/DIS 8601 standard format. The letter "Z", indicating the civil time zone at Greenwich (i.e., GMT), should never be appended to a UTC time. The relationship between UTC and GMT has varied historically and with observer context. Note that in PDS data sets created under earlier versions of the Standards, an appended Z is taken as indicating UTC.

The START_TIME and STOP_TIME data elements required in data product labels and catalog files are in UTC. For data collected by spacecraft-mounted instruments, the date/ time must be a time that corresponds to spacecraft event time. For data collected by instruments not located on a spacecraft, this time shall be an earth-based event time.

Adoption of UTC (rather than spacecraft-clock-count, for example) as the standard facilitates comparison of data from a particular spacecraft or ground-based facility with data from other sources.

13.3.2 Native Times

Times in any format other than the ISO/DIS 8601 format described above are considered to be in a format native to the data set, and thus native times. Data preparers should consult a PDS data engineer for assistance in selecting appropriate PDS data elements for expressing native times.

Examples of quantities that may be expressed in native time formats are included below.

13.3.2.1 Spacecraft Clock Count (sclk)

There is one native time of particular interest, however, which has specific keywords associated with it. The spacecraft clock reading (that is, the count) often provides the essential timing information for a space-based observation. Therefore, the elements SPACECRAFT_CLOCK_START_COUNT and SPACECRAFT_CLOCK_STOP_COUNT are required in labels describing space-based data. This value is formatted as a string to preserve precision.

13.3.2.2 Ephemeris Time

Need to talk to Chuck and/or Boris to determine if they want to say anything here.

13.3.2.3 Relative Time

13.3.2.4 Local (Solar) Time

The following text is pulled from the PDS3 PSDD and needs to be reviewed for inclusion in this document.

13.3.2.4.1 Local True Solar Time

The LOCAL_TRUE_SOLAR_TIME element describes the local true solar time, or LTST. It is one of two types of solar time used to express the time of day at a point on the surface of a planetary body. LTST is measured relative to the true position of the Sun as seen from a point on the planet's surface. The coordinate system used to define LTST has its origin at the center of the planet. Its Zaxis is the north pole vector (or spin axis) of the planet. The X-axis is chosen to point in the direction of the vernal equinox of the planet's orbit. (The vernal or autumnal equinox vectors are found by searching the planetary ephemeris for those times when the vector from the planet's center to the Sun is perpendicular to the planet's north pole vector. The vernal equinox is the time when the Sun appears to rise above the planet's equator.) Positions of points in this frame can be expressed as a radius and areocentric 'right ascension' and 'declination' angles. The areocentric right ascension angle, or ARA, is measured positive eastward in the equatorial plane from the vernal equinox vector to the intersection of the meridian containing the point with the equator. Similarly, the areocentric declination is the angle between the equatorial plane and the vector to the point. LTST is a function of the difference between the ARAs of the vectors to the Sun and to the point on the planet's surface. Specifically,

LTST = (a(P) - a(TS)) * (24 / 360) + 12

where,

LTST = the local true solar time in true solar hours a(P) = ARA of the point on the planet's surface in deg a(TS) = ARA of the true sun in deg

The conversion factor of 24/360 is applied to transform the angular measure in decimal degrees into hours-minutes-seconds of arc. This standard representation divides 360 degrees into 24 hours, each hour into 60 minutes, and each minute into 60 seconds of arc. The hours, minutes, and seconds of arc are called 'true solar' hours, minutes, and seconds when used to measure LTST. The constant offset of 12 hours is added to the difference in ARAs to place local noon (12:00:00 in hours, minutes, seconds) at the point where the Sun is directly overhead; at this time, the ARA of the true sun is the same as that of the surface point so that a(P) - a(TS) = 0. The use of 'true solar' time units can be extended to define a true solar day as 24 true solar hours. Due to the eccentricity of planetary orbits and the inclination of orbital planes to equatorial planes (obliquity), the Sun does not move at a uniform rate over the course of a planetary year. Consequently, the number of SI seconds in a true solar day, hour, minute or second is not constant. (Definition adapted from [VAUGHAN1995].)

13.3.2.4.2 Local Mean Solar Time

The desire to work with solar days, hours, minutes, and seconds of uniform length led to the concept of the fictitious mean Sun or FMS. The FMS is defined as a point that moves on the celestial equator of a planetary body at a constant rate that represents the average mean motion of the Sun over a planetary year. Local mean solar time, or LMST, is defined, by analogy with LTST, as the difference between the areocentric right ascensions of a point on the surface and of the FMS. The difference between LTST and LMST varies over time. The length of a mean solar day is constant and can be computed from the mean motion of the FMS and the rotation rate of a planet. The mean solar day is also called a 'sol'. Mean solar hours, minutes, and seconds are defined in the same way as the true solar units.

The acceptable range of values for local_mean_solar_time is '00:00:00.000' to '23:59:59.999'.

(Definition adapted from [VAUGHAN1995].)

13.4 Midnight and Leap Seconds

The ISO/DIS 8601 standard for representation of midnight and leap seconds are also used in PDS time fields.

13.4.1 Midnight

Midnight may be indicated in one of two ways: as 00:00:00 or 24:00:00. The usual precision modifications apply as well i.e. 24:00 is also recognized as midnight.

The 00:00:00 notation is used to indicate midnight at the beginning of a date. 24:00:00 is used to indicate midnight at the end of a date. So, for example, the following two date/time strings refer to precisely the same moment:

2007-04-07T24:00:00 = 2007-04-08T00:00:00

When the hours field has the value 24, any and all subsequent time fields must be zero.

13.4.2 Leap Seconds

Leap seconds may be positive or negative, but in either case are always applied at the end of the day in question. A positive leap second is indicated with a time value of 23:59:60. A negative leap second is indicated by the omission of the time 23:59:59. That is, on the day of a negative leap second, the sequence leading through midnight is:

23:59:57
23:59:58
00:00:00
00:00:01

And on the day of a positive leap second, the sequence through midnight is:

23:59:58
23:59:59
23:59:60
00:00:00
00:00:01

Note that the only time when the seconds value of a time string may contain the value 60 is when this represents a positive leap second.

Chapter 14

Units of Measurement

The uniform use of units of measure facilitates broad catalog searches across archive systems.The PDS standard system for units, where applicable, is the Systeme Internationale d'Unites (SI). The default units for data elements in the PSDD are determined as each element is defined and added to the dictionary. Specific unit definitions are also included in the PSDD.

The PDS allows exceptions to the SI unit requirement when common usage conflicts with the SI standard (e.g., angles which are measured in degrees rather than radians).

Both singular and plural unit names, as well as unit symbols, are allowed. The double asterisk (**) is used, rather than the caret (ˆ), to indicate exponentiation. When the units associated with a value of a PDS element are not the same as the default units specified in the PSDD (or when explicit units are preferred), a unit expression is used with the value. These unit expressions are enclosed in angular brackets (¡ ¿) and follow the value to which they apply.

Update to indicate implementation using attributes.

Examples Needs to be udpated

EXPOSURE_DURATION = 10 seconds
DECLINATION = -14.2756 degrees
MASS = 123 kg
MASS_DENSITY = 123 g/cm^3
MAP_RESOLUTION = 123 pixel/degree
MAP_SCALE = 123 km/pixel

Note that in the above example, MASS_DENSITY is not expressed in the SI default unit of measurement for density (kg/m^3).

PDS recommends (in order of preference) that measurements be expressed using the default SI units of measurements, as defined in the following paragraphs. If it is not desirable to use the default SI unit of measurement, then the unit of measurement should be expressed using the SI nomenclature defined in the following paragraphs. If a unit of measurement is not defined by the SI standard, then a unit of measurement can be derived (e.g., pixels per degree, kilometers per pixel, etc.).

14.1 SI Units

The following summary of SI unit information is extracted from The International System of Units.

Base units As the system is currently used, there are seven fundamental SI units, termed base units:

QUANTITY	NAME OF UNIT	SYMBOL
length	meter	m
mass	kilogram	kg
time	second	s
electric current	ampere	A
thermodynamic temperature	kelvin	K
amount of substance	mole	mol
luminous intensity	candela	cd

SI units are all written in mixed case; symbols are also mixed case except for those derived from proper names. No periods are used in any of the symbols in the international system.

Derived units In addition to the base units of the system, a host of derived units, which stem from the base units, are also employed. One class of these is formed by adding a prefix, representing a power of ten, to the base unit. For example, a kilometer is equal to 1,000 meters, and a millisecond is .001 (that is, 1/1,000) second. The prefixes in current use are as follows:

SI PREFIXES

Factor	Prefix	Symbol	Factor	Prefix	Symbol
10^{18}	exa	E	10^{-1}	deci	d
10^{15}	peta	P	10^{-2}	centi	c
10^{12}	tera	T	10^{-3}	milli	m
10^{9}	giga	G	10^{-6}	micro	
10^{6}	mega	M	10^{-9}	nano	n

10^3	kilo	k	10^{-12}	pico	p
10^2	hecto	h	10^{-15}	femto	f
10^1	deka	da	10^{-18}	atto	a

Note that the kilogram (rather than the gram) was selected as the base unit for mass for historical reasons. Notwithstanding, the gram is the basis for creating mass units by addition of prefixes.

Another class of derived units consists of powers of base units and of base units in algebraic relationships. Some of the more familiar of these are the following:

QUANTITY	**NAME OF UNIT**	**SYMBOL**
area	square meter	m^2
volume	cubic meter	m^3
density	kilogram per cubic meter	kg/m^3
velocity	meter per second	m/s
angular velocity	radian per second	rad/s
acceleration	meter per second squared	m/s^2
angular acceleration	radian per second squared	rad/s^2
kinematic viscosity	square meter per second	m^2/s
dynamic viscosity	newton-second per square meter	$N * s/m^2$
luminance	candela per square meter	cd/m^2
wave number	1 per meter	m^{-1}
activity (of a radioactive source)	1 per second	s^{-1}

Many derived SI units have names of their own:

QUANTITY	**NAME OF UNIT**	**SYMBOL**	**EQUIVALENT**
frequency	hertz	Hz	s^-1
force	newton	N	$kg * m/s^2$
pressure (mechanical stress)	pascal	Pa	N/m^2
work, energy, quantity of heat	joule	J	$N * m$
power	watt	W	J/s
quantity of electricity potential difference	coulomb	C	$A * s$
electromotive force	volt	V	W/A
electrical resistance	ohm		V/A
capacitance	farad	F	$A * s/V$
magnetic flux	weber	Wb	$V * s$
inductance	henry	H	$V * s/A$
magnetic flux density	tesla	T	Wb/m^2

luminous flux	lumen	lm	$cd * sr$
illuminance	lux	lx	lm/m^2

Supplementary units are as follows:

QUANTITY	**NAME OF UNIT**	**SYMBOL**
plane angle	radian	rad
solid angle	steradian	sr

Use of figures with SI units In the international system it is considered preferable to use only numbers between 0.1 and 1,000 in expressing the quantity associated with any SI unit. Thus the quantity 12,000 meters is expressed as 12 km, not 12,000 m. So too, 0.003 cubic centimeters is preferably written 3 mm3, not 0.003 cm3.

Appendices

Index